Generative Adversarial Networks

Doc.dr.sc. Marko Subašić

GANs

- Primarna namjena je generiranje uzoraka
- Ne procjenjujemo p(**x**)
- Ne traži uzorkovanje
  - Kod VAE smo aproksimativno uzeli jedan uzorak od **z** iako smo trebali uzeti više njih
- Asimptotski konzistentan za razliku od VAE
- Često generira vizualno najbolje uzorke
  - Nije jasno zbog čega

Osnovna ideja

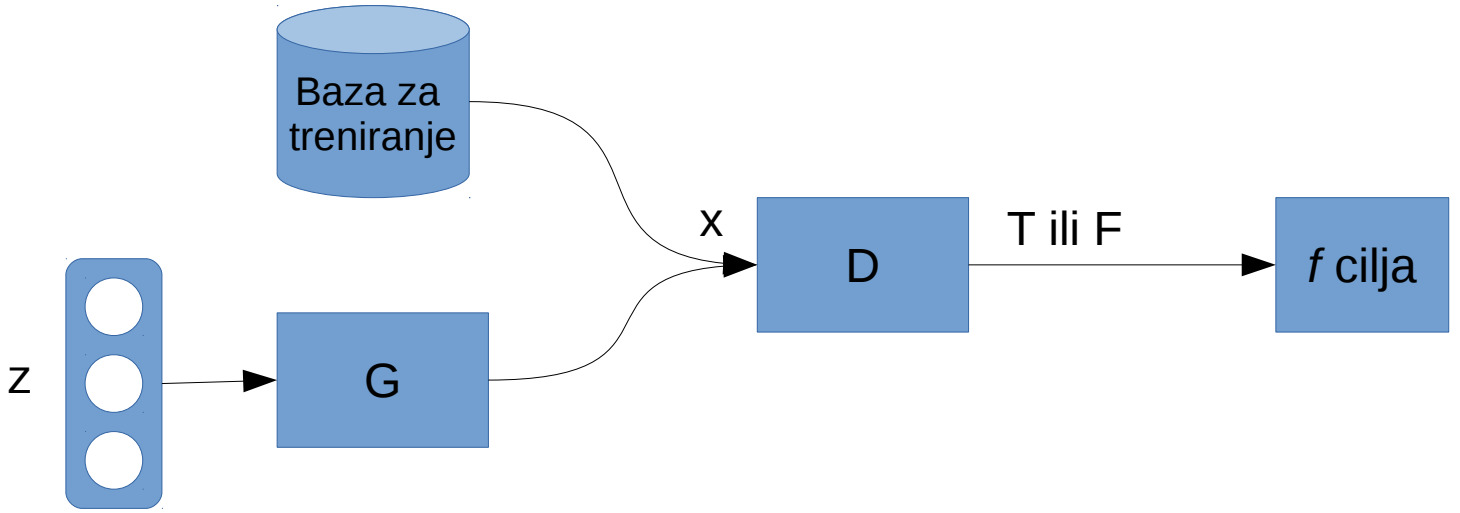
- Dvije mreže:

1) Generator (G): generira umjetne uzorke
2) Diskriminator (D): pokušava otkriti umjetne uzorke

- Dvije mreže su protivnici (adversaries)
  - Jedna pokušava prevariti drugu
  - Druga pokušava detektirati prevaru
  - Njihovo natjecanje tjera ih da budu sve bolji i bolji u svojim nastojanjima
  - U konačnici generator generira "savršene" umjetne uzorke koje diskriminator ne može razlikovati od pravih uzoraka

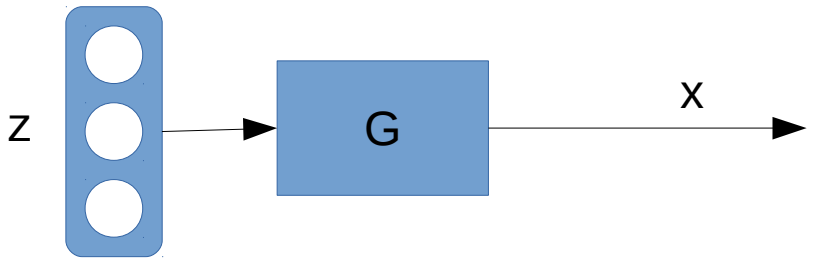
Osnovna ideja

- Dvostruka povratna veza
  - Diskriminator sa uzorcima iz skupa za treniranje
  - Generator sa diskriminatorom
- Može dovesti do nestabilnosti



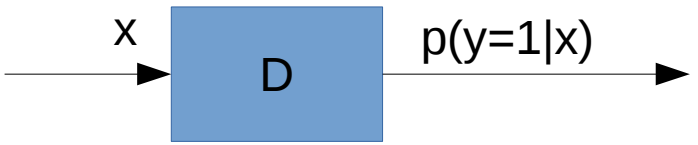
Generator

- Želimo da slučajno generira različite uvjerljive uzorke
  - Da bi to postigli, negdje u mreži treba biti ugrađena "slučajnost"
  - Ulaz (z) će biti slučajni šum s nekom odabranom disitribucijom p(z)
  - Ne pokušavamo pronaći p(z|x) – jednostavnost!
  - Generator mora biti diferancijabilan



Diskriminator

- Treniran da bi razlikovao stvarne uzorke iz skupa za treniranje od umjetnih uzoraka koje proizvodi generator
  - Binarna klasifikacija
    - Klasa skupa za treniranje y=1, klasa umjetnih uzoraka y=0
    - Klasifikacija p(y=1|x)
    - Učenje pod nadzorom
  - On je pomoćni alat za poboljšanje generatora
    - Može se odbaciti, ali i ne mora...



Funkcija cilja

- Generator $G(z;\theta_g)$
  - θ_g su parametri generatora
  - Mapira ulazni prostor skrivene varijable z u prostor podataka
- Diskriminator $D(x;\theta_d)$
  - θ_d su parametri diskriminatora
  - Skalarna funkcija podataka koja procjenjuje vjerojatnost da ulazni podatak nije umjetno generiran

Funkcija cilja

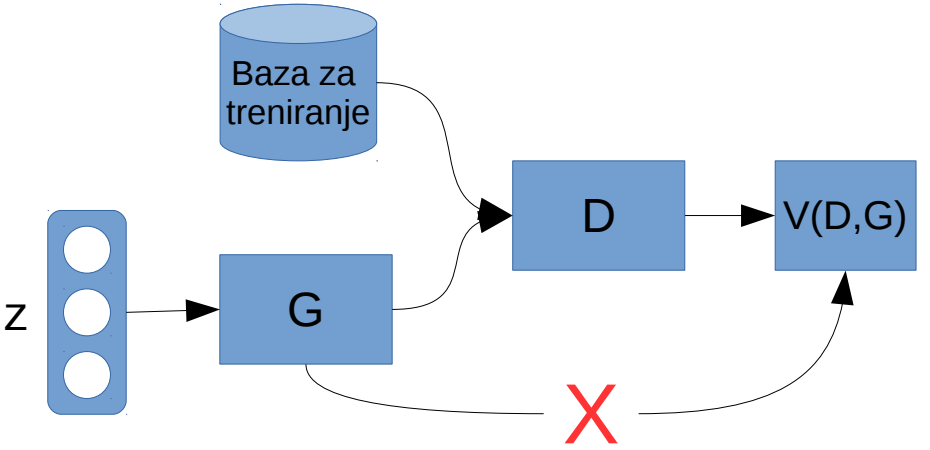
- Treniranjem se optimizara funkcija cilja s obzirom na parametre obje mreže

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Procijenjena log vjerojatnost da je stvarni uzorci nisu umjetni

Procijenjena log vjerojatnost da generirani podaci jesu umjetni

- Minimax igra
  - Dvije mreže su suprotstavljene
  - Traži se ekvilibrij igre u sedlu – Nash-ev ekvilibrij



Što možemo reći da GAN radi

1) Automatizira testiranje i poboljšanje generatora
   - Bitno pitanje kod generativnih modela je kako ocijeniti kvalitetu novih generiranih uzoraka
     - Nema ih se sa čime usporediti
   - Diskriminator pokušava pronaći razlike između stvarnih slika i novih generiranih slika
     - Učenje pod nadzorom
     - Ako je uspješan (bolji od slučajnog odabira), onda generator ima mjesta za poboljšanje

Što možemo reći da GAN radi

2) Adaptivno treniranje

- Kako generator postaje sve bolji, diskriminator se tome prilagođava i pronalazi nove greške (razlika) koje generator radi
- Kako diskriminator postaje sve bolji, generator pronalazi nove načine da ga prevari

Što možemo reći da GAN radi

3) Minimizira divergenciju
   - Treniranje GAN-a je ekvivalentno minimizaciji Jensen-Shannonove divergencije

Koko se trenira GAN?

- Nije zadano...
- Zašto ne backprop?
- Funkcija cilja se lako evaluira

$$V(D,G)=E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

- Naizmjenično treniranje generatora i diskriminatora

Treniranje

- Osvježavanje parametara obje mreže

$$\theta_g^{t+1} = \theta_g^t + \eta \nabla_{\theta_g} V(\theta_g \theta_d)$$
$$\theta_d^{t+1} = \theta_d^t - \eta \nabla_{\theta_d} V(\theta_g \theta_d)$$

Algoritam treniranja

for broj iteracija **do**

for k koraka **do**

Uzmi mini grupu od *m* uzoraka *z* iz distribcije $p_g(z)$

Uzmi mini grupu od *m* uzoraka *x* iz distribcije $p_{data}(x)$

Osvježi diskriminator prema

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

Uzmi mini grupu od *m* uzoraka *z* iz distribcije $p_g(z)$

Osvježi generator prema

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

Konvergencija

- Što je optimalni diskriminator?

$$V(G,D)=\int_x p_{data}(x)\log(D(x))\,dx+\int_z p_Z(z)\log(1-D(G(z)))\,dz$$
$$=\int_x p_{data}(x)\log(D(x))+p_g(x)\log(1-D(x))\,dx$$
$$\leqslant\int_x \max_y \left[p_{data}(x)\log(y)+p_g(x)\log(1-y)\right]dx$$

$$f(y)=a\log y+b\log(1-y)$$
$$f'(y)=0\rightarrow\frac{a}{y}-\frac{b}{1-y}=0\rightarrow y=\frac{a}{a+b}$$

$$D_G=\frac{p_{data}}{p_{data}+p_g}$$

Konvergencija

- Što je optimalni diskriminator uz optimalni generator?

$$p_{data} = p_g \rightarrow D_G^* = \frac{p_{data}}{p_{data} + p_g} = \frac{1}{2}$$

  - Daje 1/2 za sve uzorke bez obzira odakle dolaze

Konvergencija

- A koji bi bio optimalni generator?

$$\min_G \max_D V(D_G^*, G) = C(G) = E_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}\left[\log \frac{p_{data}}{p_{data}+p_g}\right] + E_{\boldsymbol{x} \sim p_g(\boldsymbol{x})}\left[\log \frac{p_g}{p_{data}+p_g}\right]$$

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

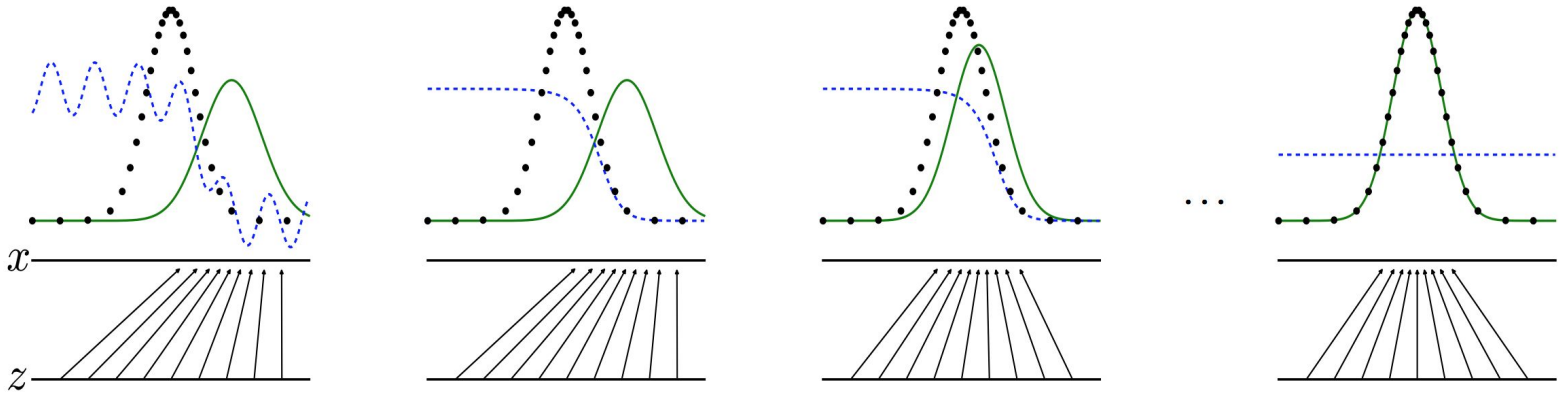
- KL divergencija je uvijek pozitivna

$$C(G) = -\log 4 + 2\,JDS(p_{data} | p_g)$$

- Jenson-Shannon divergencija
  - jednaka 0 samo hada je $p_{data} = p_g$

Konvergencija

- Uz dovoljne kapacitete D i G
  - D će konvergirati u svoj optimum D_G^* za zadani G
    - D pokušava procijeniti omjer vjerojatnosti pomoću učenja pod nadzorom
  - G primiče p_g prema p_{data}
    - Izbor generatora ograničava p_g
    - Za višeslojni perceptron sve dobro funkcionira



Intutitvno

- Da bi generator bio dobar
  - Diskriminator mora biti približno idealan
  - Diskriminator je obično složeniji

Što je problematično

- Nejasan kriterij zaustavljanja
- Istovremeno slijeđenje gradijenta dviju mreža može dovesti do oscilacija ili nestabilnosti
  - Nemogućnost konvergencije
- Nema eksplicitne reprezentacije $p_g(x)$
- Teška usporedba s drugim generativnim modelima

Što je problematično

- Treniranje može dovesti do pamćenja uzoraka iz skupa za treniranje
- Teško je generirati skrivene varijable z iz ulaznih uzoraka x
- Mode collapse – generiranje uvijek iste slike
  - Nije uvijek identična, ali vrlo slična
  - Postoje razlike ali su male

Razlika u odnosu na VAE

- VAE indirektno optimizira funkciju cilja p(x)
- GAN direktno minimizira funkciju cilja
  - Diskriminator određuje/uči funkciju cilja

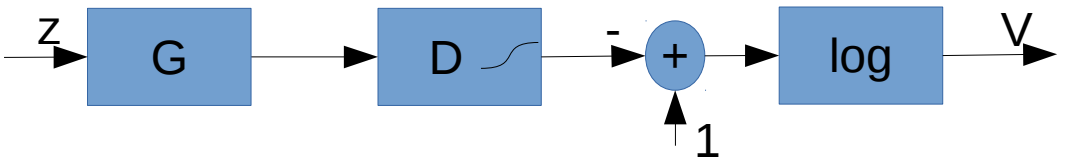
Neke varijacije

- Kako bi se izbjegli problemi GAN mreža

Varijacija funkcije cilja za G

- Nema garancija da će paralelni backpropagation konvergirati
- Postoje razni heuristički trikovi
- Modifikacija cilja kako bi se izbjeglo prerano zasićenje gradijenata ako je G loš

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



Feature matching

- Može pomoći gdje je obični GAN nestabilan
- Dodatni član funkcije cilja generatora kojim se želi postići podudarnost značajki
  - Značajkama se proglašavaju vrijednosti jednog skrivenog sloja diskriminatora - **f**(**x**)

$$\left\| E_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} \boldsymbol{f}(\boldsymbol{x}) - E_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \boldsymbol{f}(G(\boldsymbol{z})) \right\|$$

- Ideja ima smisla zato što diskriminator ima za cilj pronaći bitne značajke za razlikovanje stvarnih od umjetnih uzoraka
- Daje dobre rezultate ako se želi dobiti jači klasifikator

Minibatch discrimination

- Odgovor na mode collapse
- Ugradnja dodatnog mehanizma u diskriminator kako bi mogao detektirati mode collapse
  - Promatranje cijelog minibatch-a
  - Neki skriveni sloj diskriminatora $\mathbf{f}(\mathbf{x})$ se proširuje određenim brojem mjera varijacije unutar minibatch-a
    - Minibatch-evi koje proizvodi generator moraju oponašati varijabilnost minibatch-eva za treniranje
    - Mjere varijacije uzoraka se isto treniraju
- Daje dobre rezultate ako se teži generiranju vizualno uvjerljivi uzoraka
  - Bolje od feature matchinga

Povijesno usrednjavanje

- G i D imaju dodatni član funkcije cijene

$$\left\| \boldsymbol{\theta} - \frac{1}{t} \sum_{i=1}^{t} \boldsymbol{\theta}[i] \right\|^2$$

  - $\boldsymbol{\theta}[\,i\,]$ su parametri u prthodnom trenutku *i*
- Kažnjava se velika odstupanja težina od povijesne srednje vrijednosti
- Inspiriran rješenjima iz teorije igara
- Može pomoći kod nekonveksnih problema

Reference batch normalization

- Batch normalitzation dovodi od ovisnosti izlaza za x o ostalim uzorcima u minibatch-u
- Uvodi se referentni minibatch koji služi za normalizaciju svih minibatch-eva
  - Određuje se samo jednom
- Podrazumijeva unaprjedni prolaz kroz mrežu za trenutni minibatch i za referentni minibatch

Virtual batch normalization

- Odabere se reference batch
- Svaki element x trenutnog minibatcha se normalizira pomoću novog minibatcha koji se sastoji od x i od reference batcha

Semi supervised learning

- Izlaz D se proširuje za predikciju K klasa (K+1 izlaz)
- Diskriminator tada optimizira dvije funkcije cijene
  - Uobičajeni GAN gubitak – stvarni ili umjetni uzorak
  - Gubitak učenja pod nadzorom – procjena klase uzorka kao je ulazni uzorak iz skupa za treniranje i ako je označen
- Korištenje Feature matchinga donosi bolje rezultate za semi-supervised klasifikaciju
- Semi-supervised učenje omogućuje veću kvalitetu generiranih slika
  - Pretpostavka je da oznaka klase nosi statistiku koja je bitna za ljudsku interpretaciju

One-sided label smoothing

- Poznata metoda regularizacije

$$D_G = \frac{\alpha\, p_{data} + \beta\, p_g}{p_{data} + p_g}$$

- Cilj za pozitivnu klasifikaciju se postavlja na α (npr. 0.9) , anegativnu na β (npr. 0.1)
- p_g u brojniku predstavlja problem
  - Tamo gdje je p_{data} približno 0, a p_g velika, "pogrešno" generirani uzorci se neće približavati p_{data}
- Stoga se β postavlja na 0 – stoga "jednostrano"

One-sided label smoothing

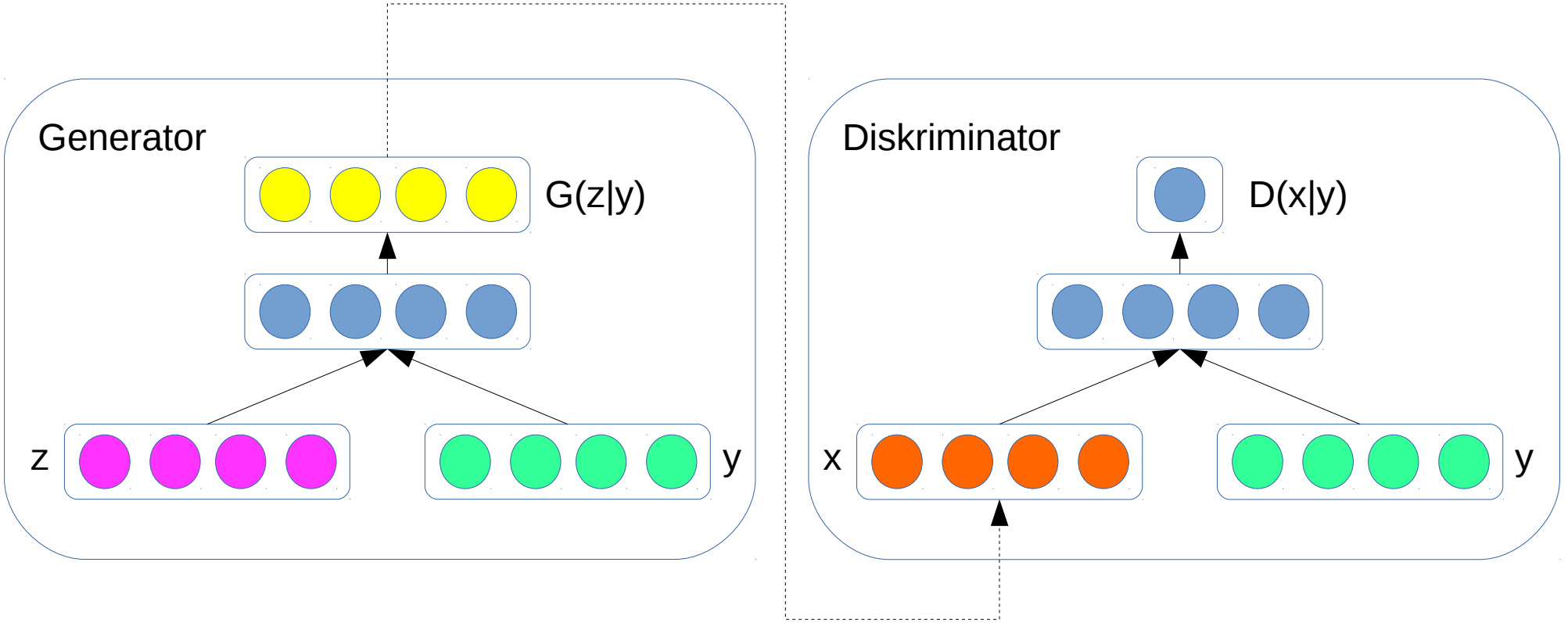
- α ne utječe na točnost nego samo na sigurnost u pozitivnu klasu
- Prednosti
  - Sprječava diskriminator da prosljeđuje veliki gradijent generatoru
  - Smanjuje se utjecaj vrlo uspješnih uzoraka

Balans G i D

- Želimo da diskriminator bude što bolji
  - D je obično veći i dublji
  - Možda je dobro osvježavati D češće nego G
- Za neke funkcije cijene može biti problem ako je D predobar, pogotovo u početku kada je G loš
  - Jedno rješenje je label smoothing

Korištenje označavanja

- Class conditional model
- Poboljšava subjektivnu kvalitetu slika



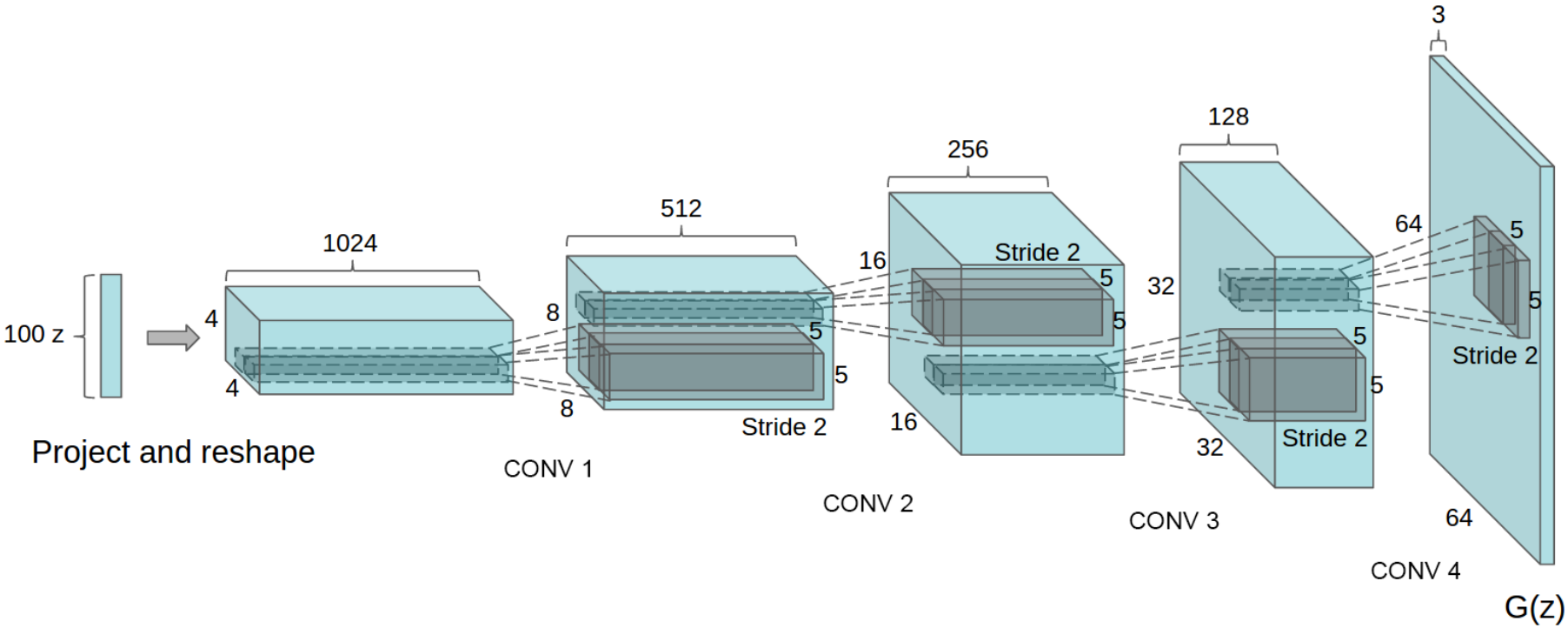
Zašto je GAN dobar

- Zašto ne generira mutne uzorke kao VAE
  - Ne zna se
  - Nije fokusiranje na drugačiju divergenciju
- Sličnost sa reinforced learning
  - Diskriminator daje ocjenu, ali se i on mijenja

DCGAN

- Deep Convolutional GAN
- Generiranje slika iz slučajnog vektora
- Sličnost sa CNN
- Stride veći od 1 (D) i manji od 1 (G)
  - Alternativa pooling slojevima
  - Veza nije predefinirana nego se optimizira
- Batch normalization (G i D)
  - Svugdje osim u izlaznom sloju generatora i ulaznom sloju diskriminatora
- Leaky ReLU aktivacija osim u izlaznom sloju (G)
- Eliminacija FC slojeva (G i D)

DCGAN - generator



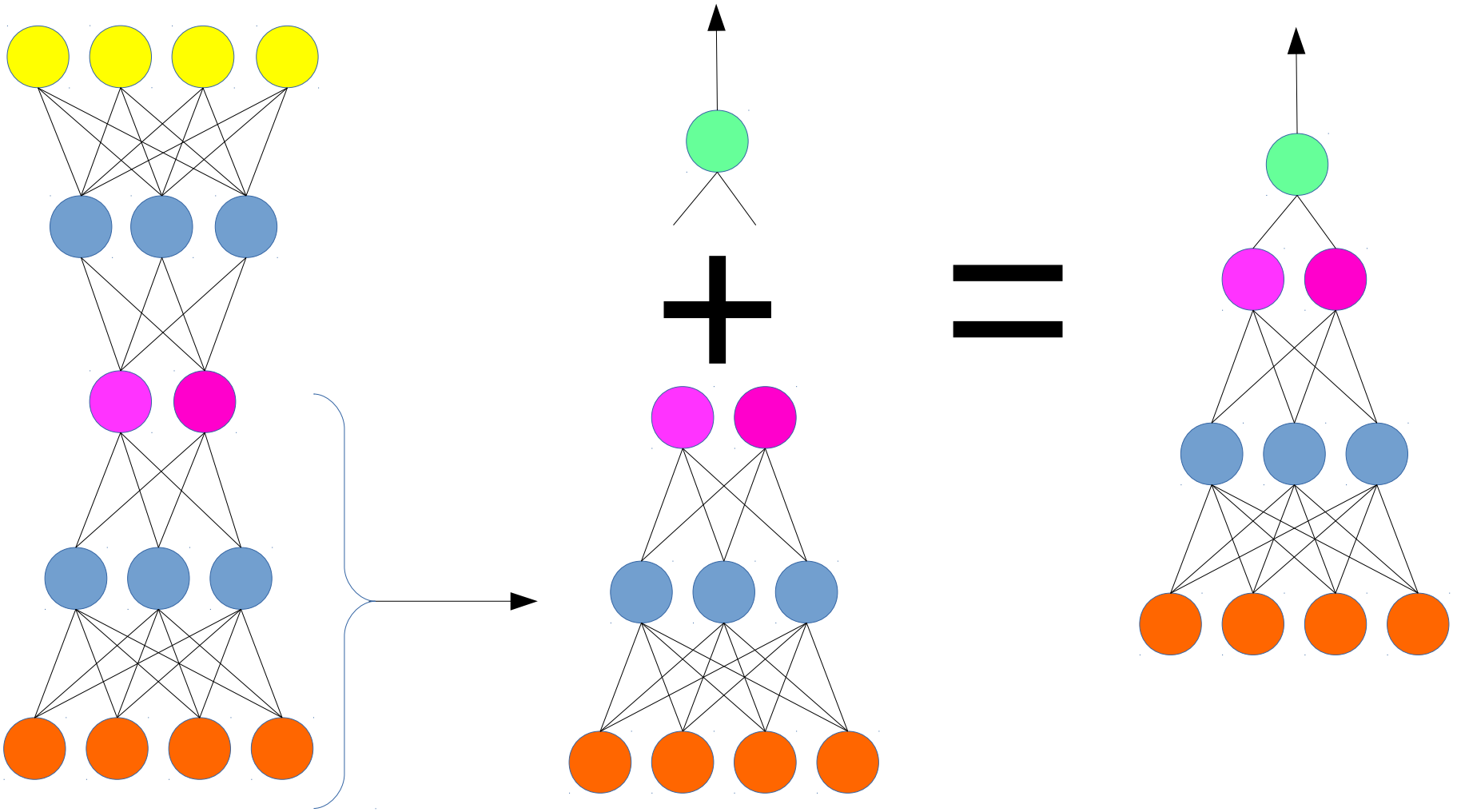
Kako koristiti GAN

- G za generiranje
- Ako se D trenira da odredi klasu uzorka (sve moguće klase plus fake klasa)
  - D se tada može koristiti za klasifikaciju

real data
gen. data
features
discriminator

Što radi VAE

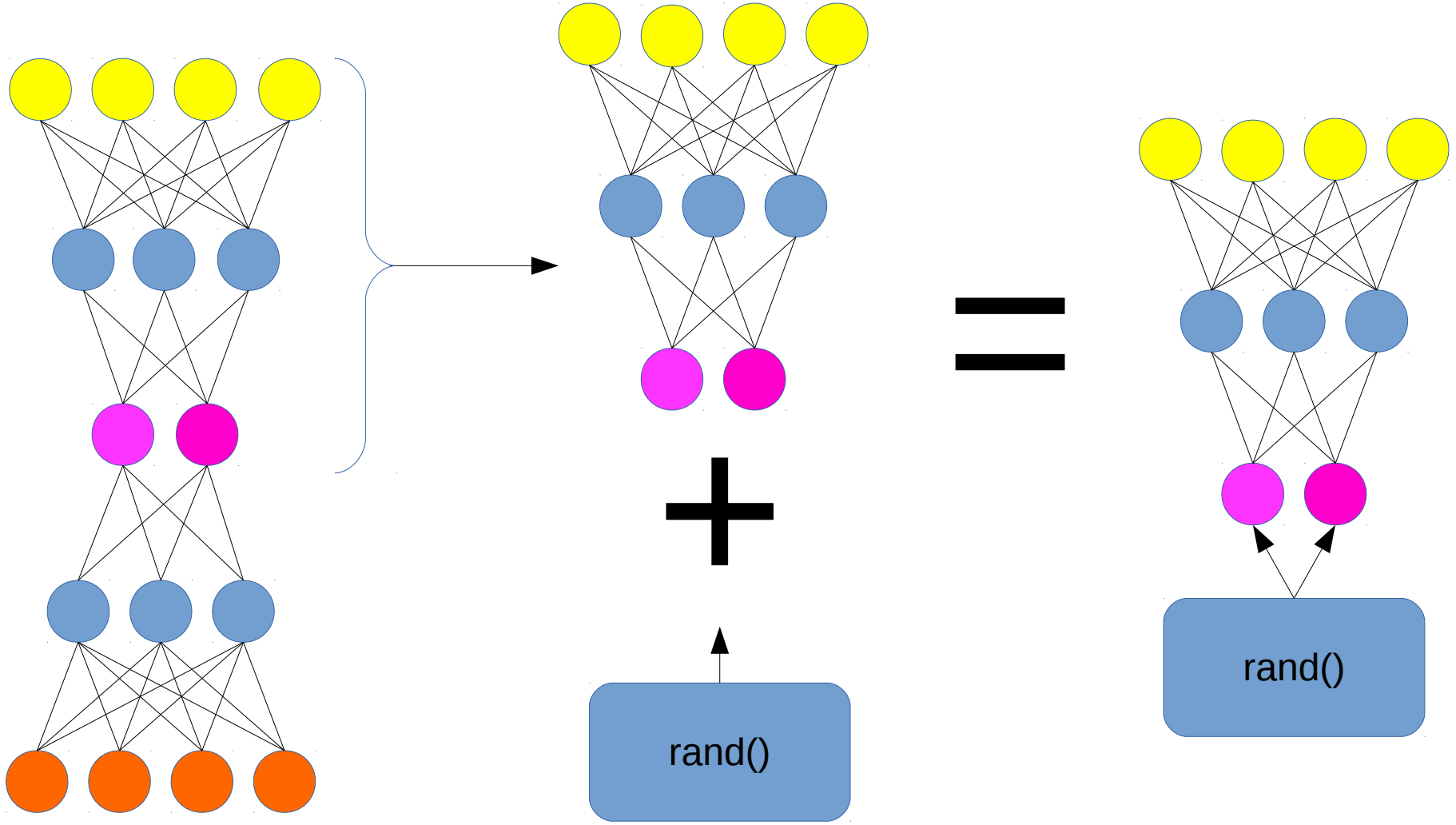
- Diskriminacija



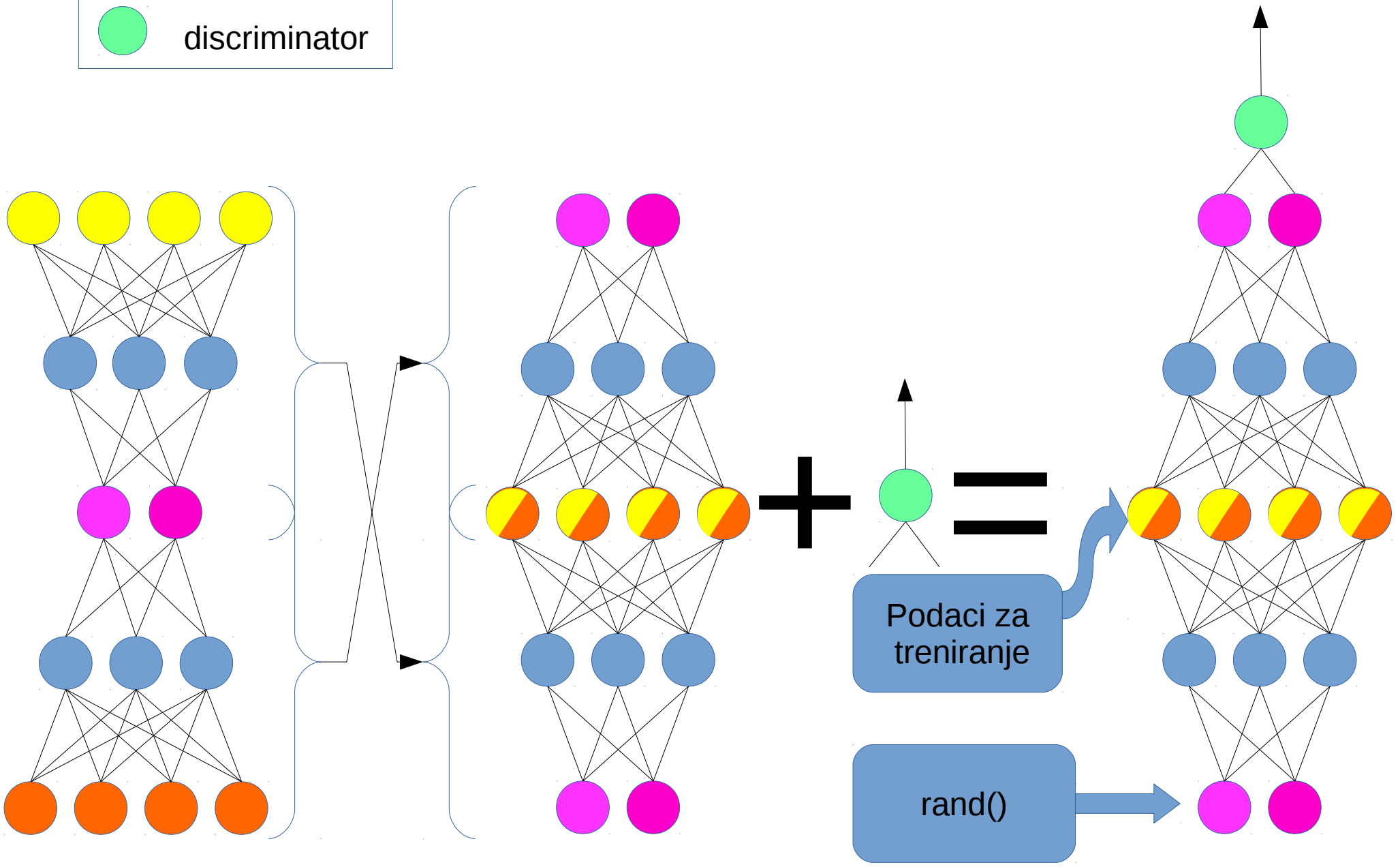
real data
gen. data
features
discriminator

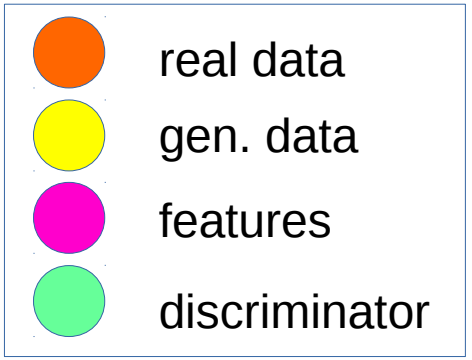
Što radi VAE

- Genereiranje uzoraka



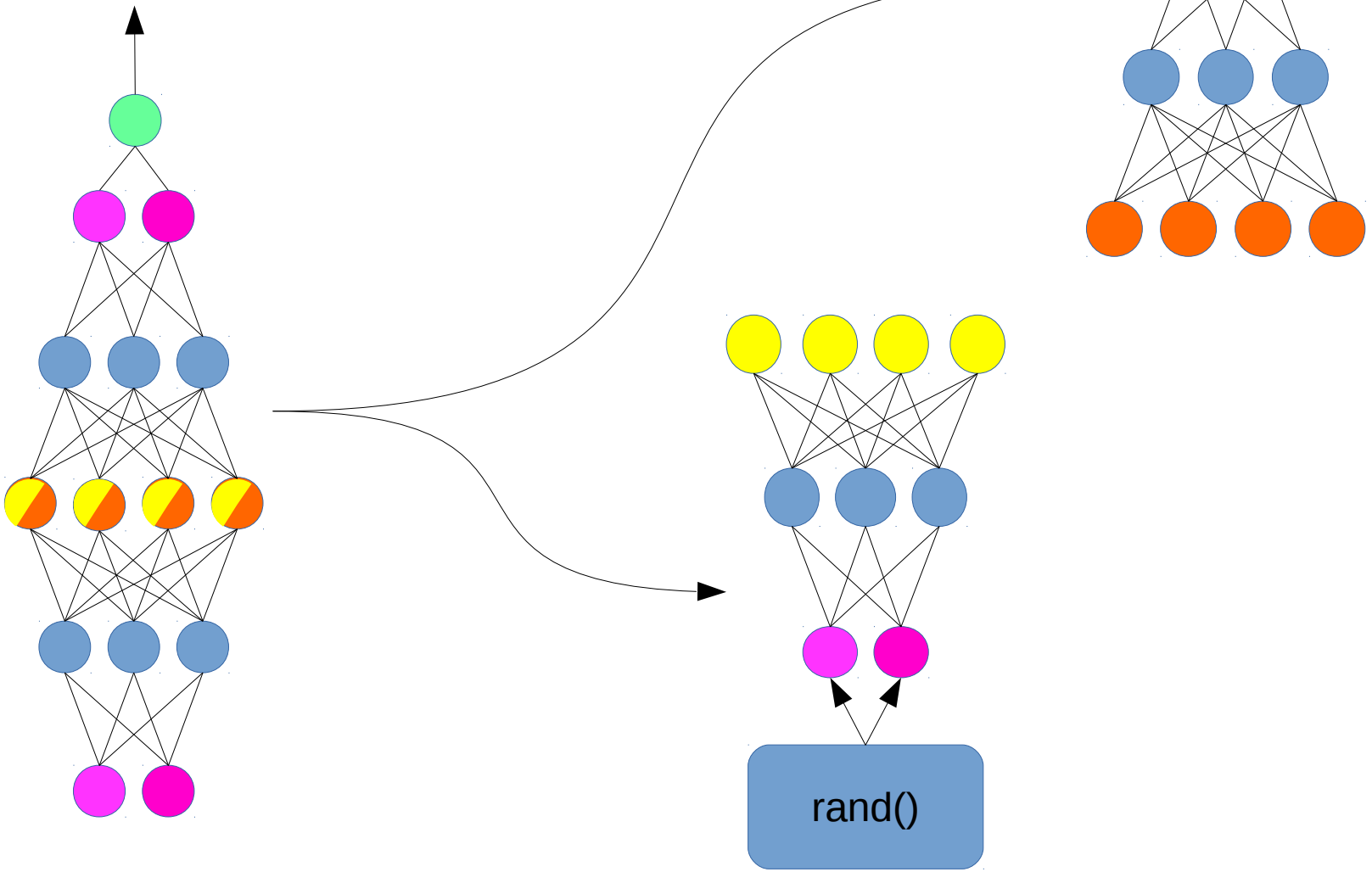
Što radi GAN

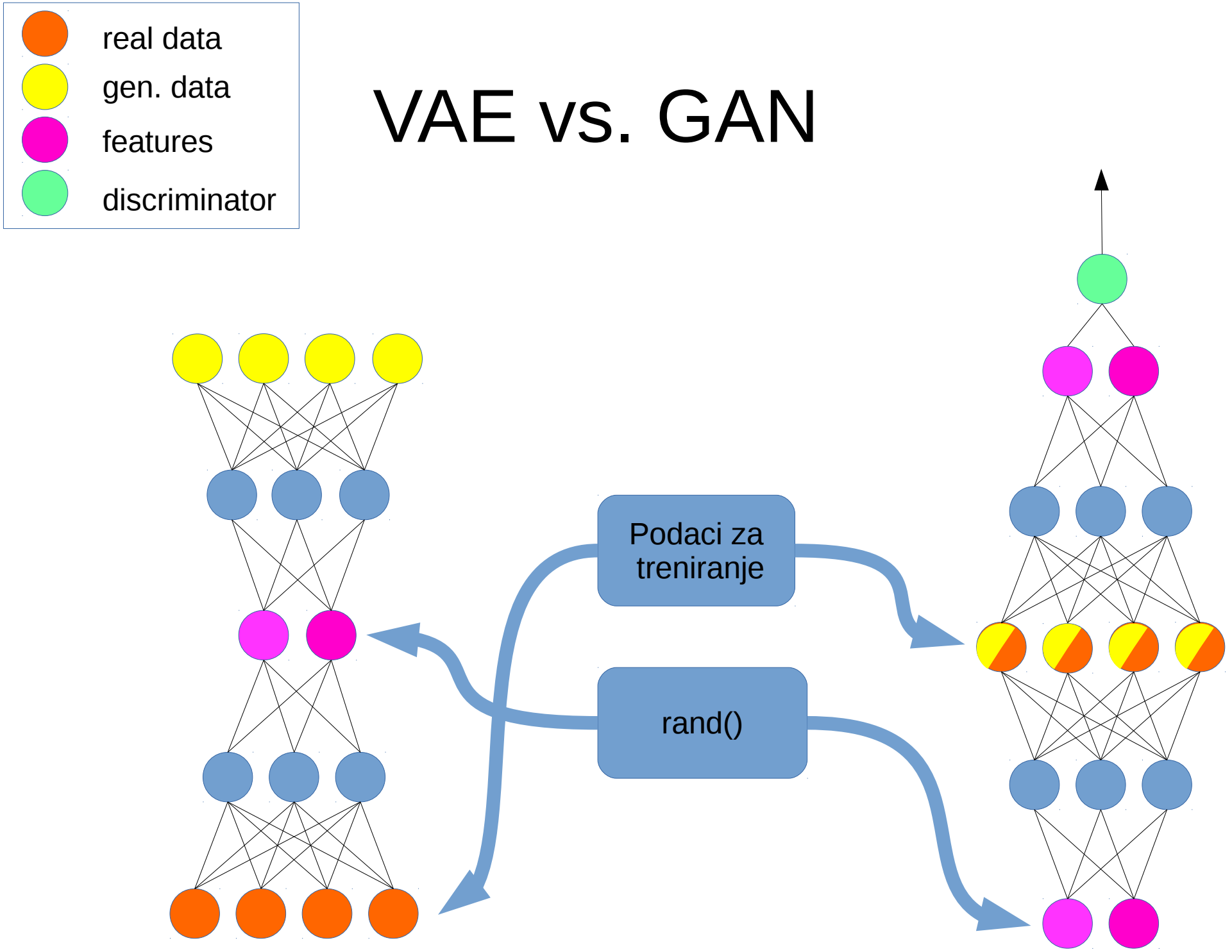

real data
gen. data
features
discriminator
Podaci za treniranje
rand()



Što radi GAN

- Diskriminacija i generiranje




real data
gen. data
features
discriminator
VAE vs. GAN
Podaci za treniranje
rand()

real data

gen. data

features

discriminator

Mogućnosti