Conditional GANs

Taking control of the output

Victor ADASCALITEI

Recap …

Recap …

Examples

Recap …

Examples

Image to image translation

Recap …

Examples

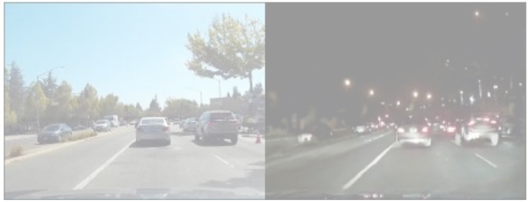


Image to image translation

Retail

Recap …

Examples

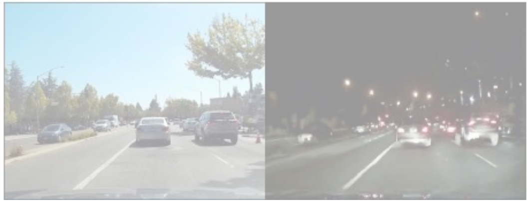


Image to image translation

Retail

Image completion

Recap …

Examples

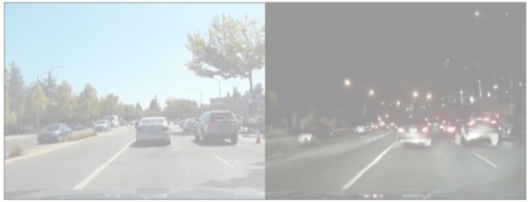


Image to image translation

Retail

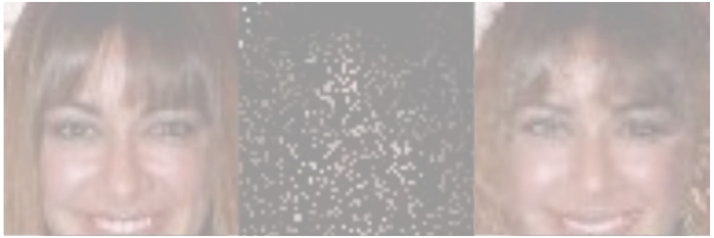
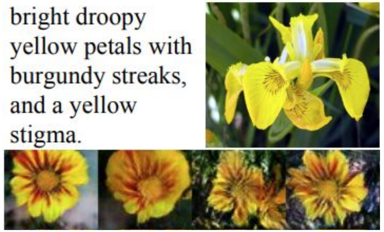


Image completion



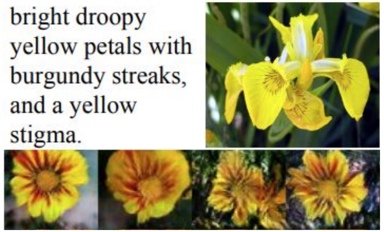
Text to image synthesis

Recap … **Some Examples**

Image to image translation [1]

Retail [2]

Image completion [3]



Text to image synthesis [4]

Recap …

More Examples

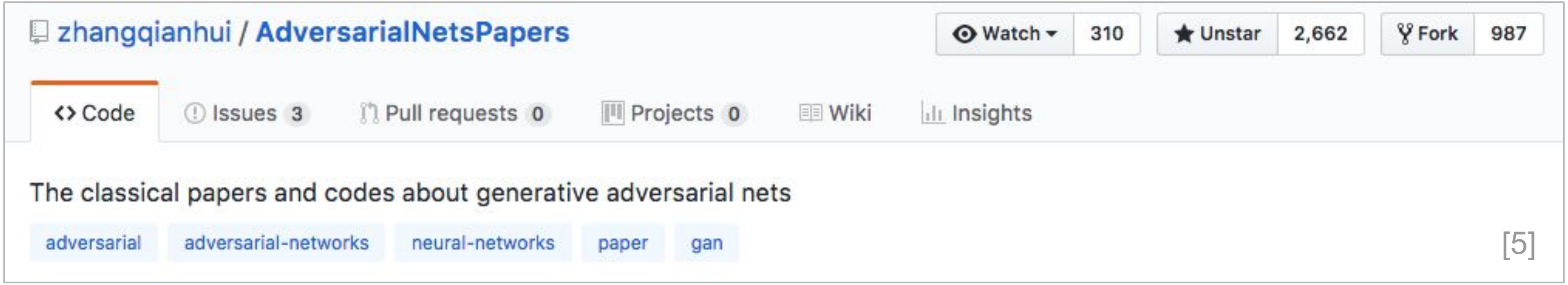
Recap …

More Examples

Semi-supervised learning
Image blending
Image inpainting
Super resolution
Semantic segmentation
Object detection
Texture synthesis and style transfer
...

Recap … More Examples

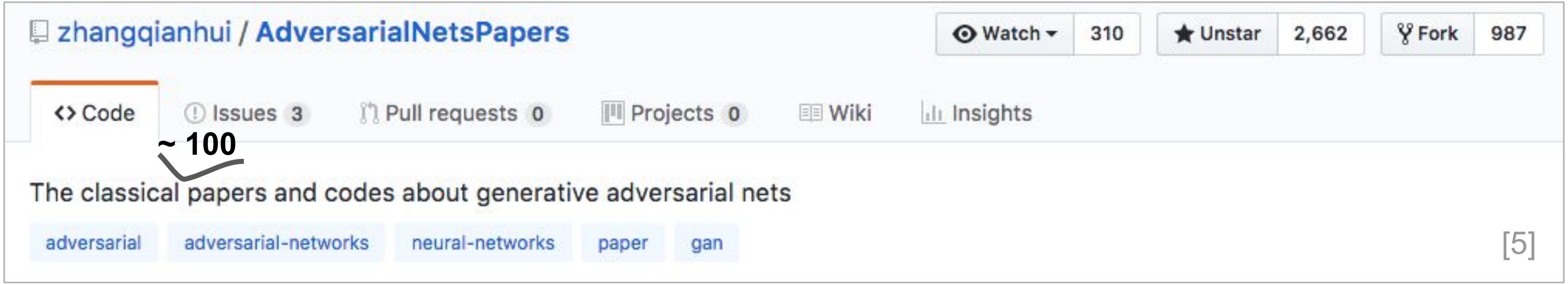
Semi-supervised learning
Image blending
Image inpainting
Super resolution
Semantic segmentation
Object detection
Texture synthesis and style transfer
...



[5]

Recap … More Examples

Semi-supervised learning
Image blending
Image inpainting
Super resolution
Semantic segmentation
Object detection
Texture synthesis and style transfer
...

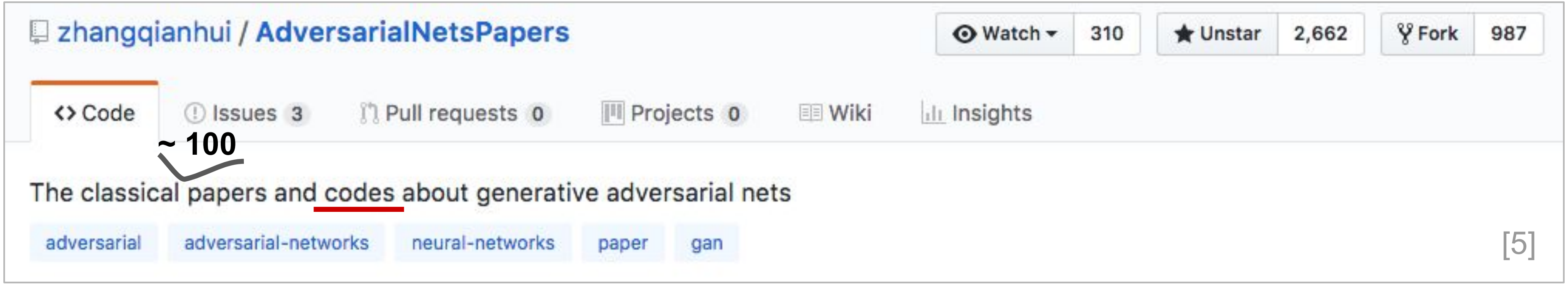


[5]

Recap …

More Examples

Semi-supervised learning
Image blending
Image inpainting
Super resolution
Semantic segmentation
Object detection
Texture synthesis and style transfer
…



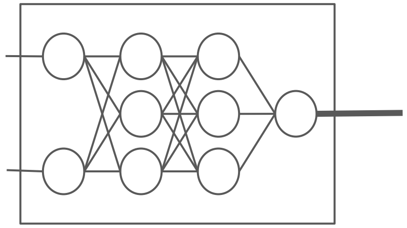
[5]

Recap …

Architecture

Recap …

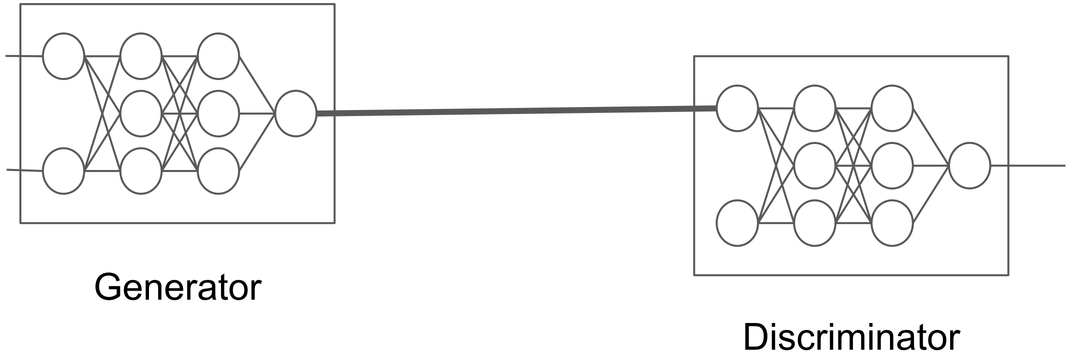
Architecture



Generator

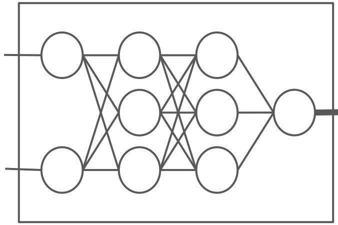
Recap …

Architecture



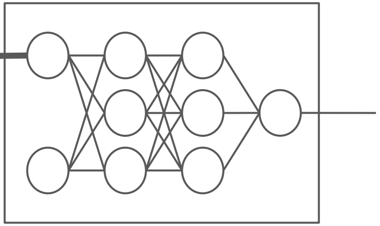
Recap …

Architecture



Generator

(Counterfeiter)

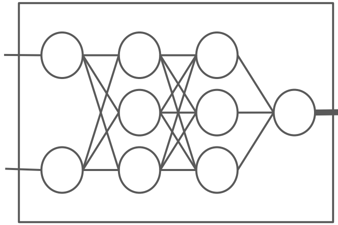


Discriminator

(Police)

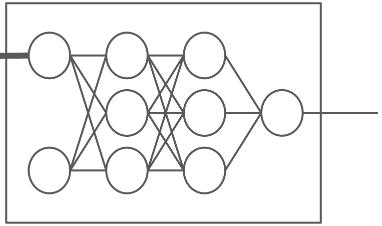
Recap …

Architecture



Generator

(Counterfeiter)



Discriminator

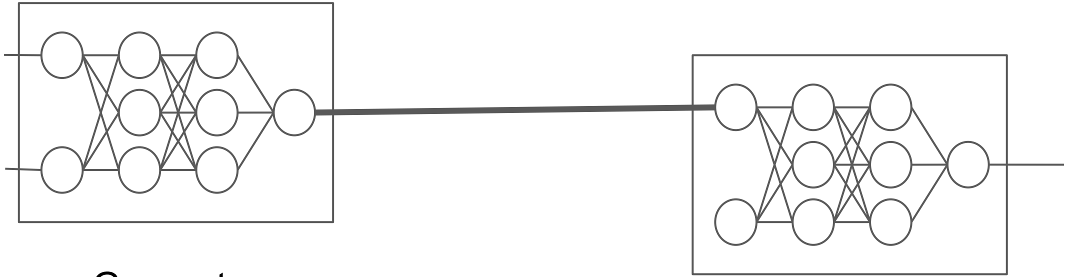
(Police)

Tries to generate data which might fool the discriminator

Tries to guess which data is authentic and which not

Recap …

Architecture



Generator

(Counterfeiter)

Discriminator

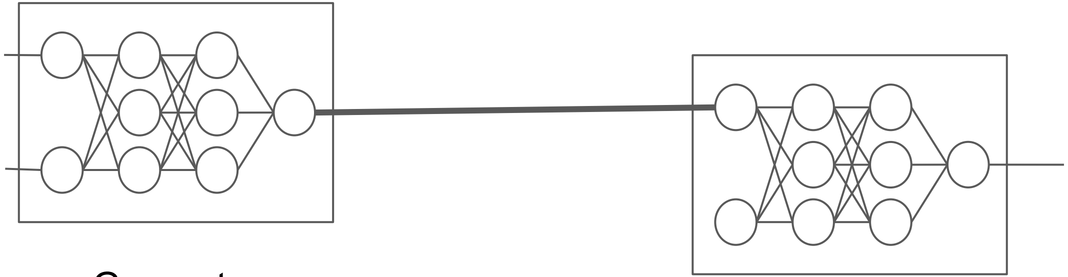
(Police)

Tries to generate data which might fool the discriminator

Tries to guess which data is authentic and which not

Recap …

Architecture



Generator

(Counterfeiter)

Discriminator

(Police)

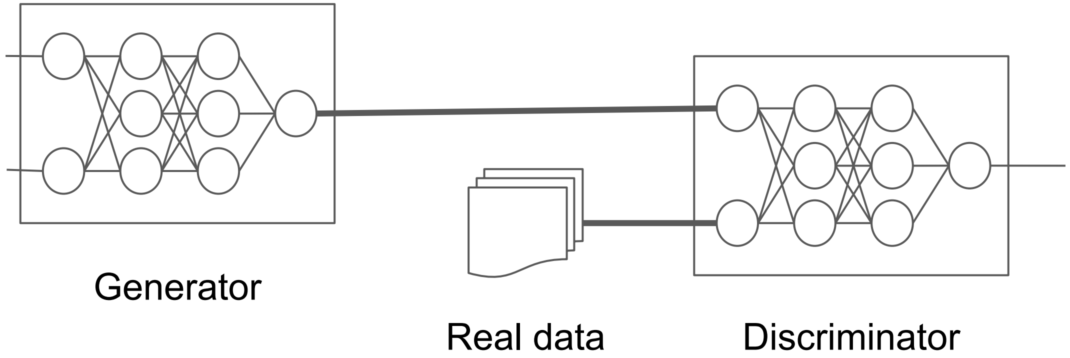
Tries to generate data which might fool the discriminator

Tries to guess which data is authentic and which not

???

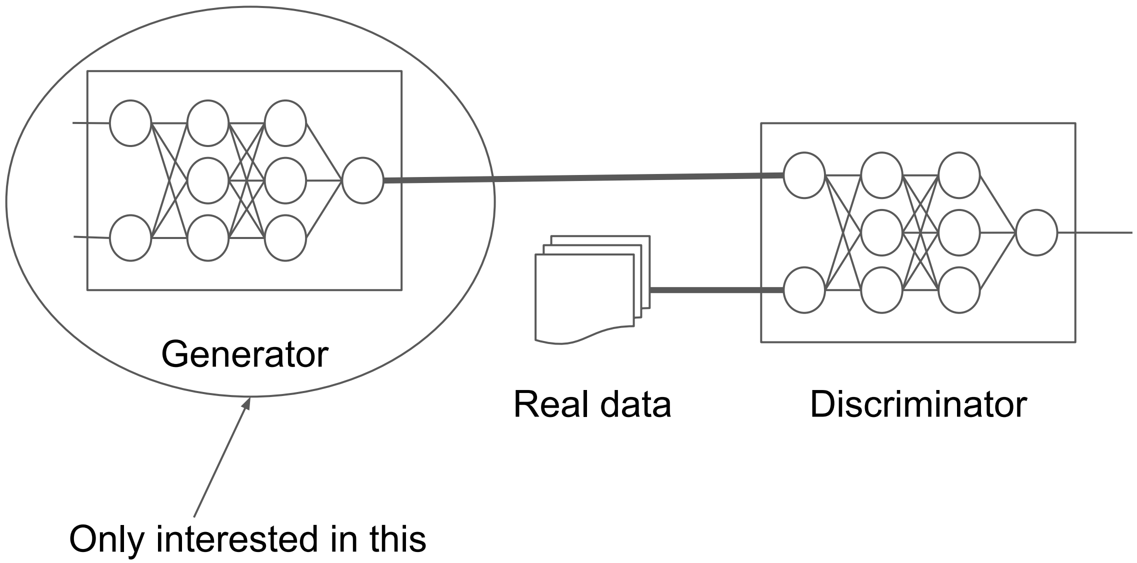
Recap …

Architecture



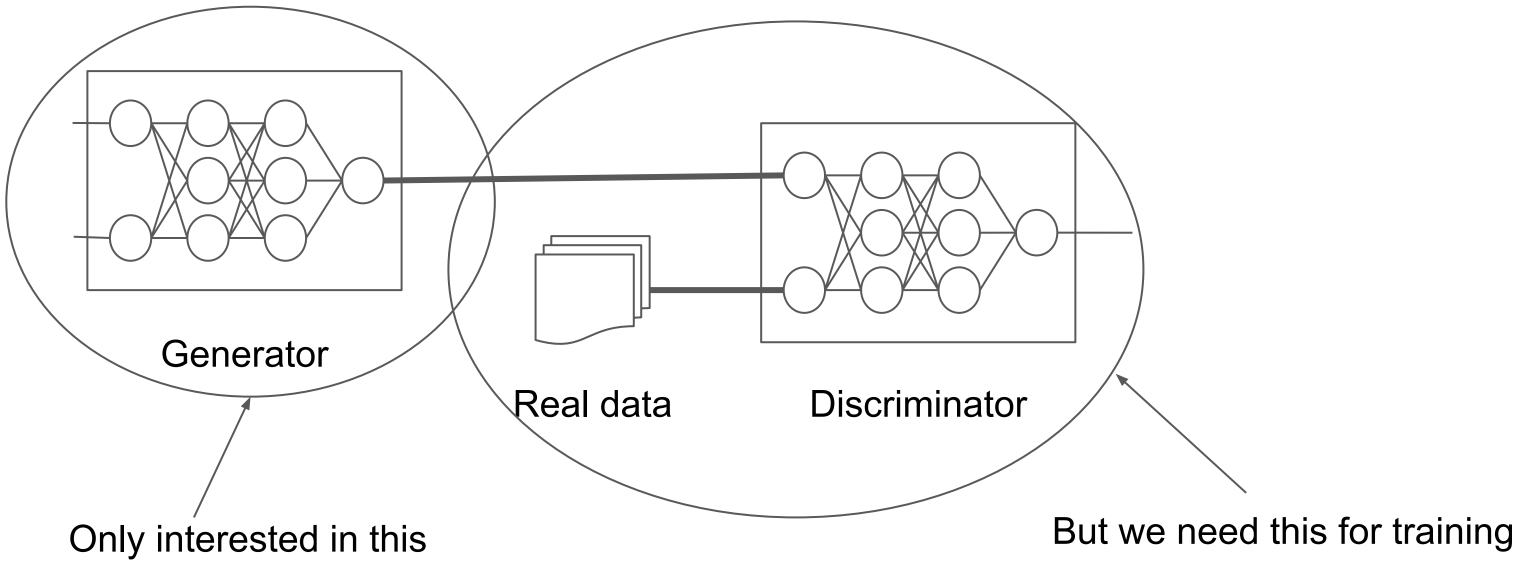
Recap …

Architecture



Recap …

Architecture

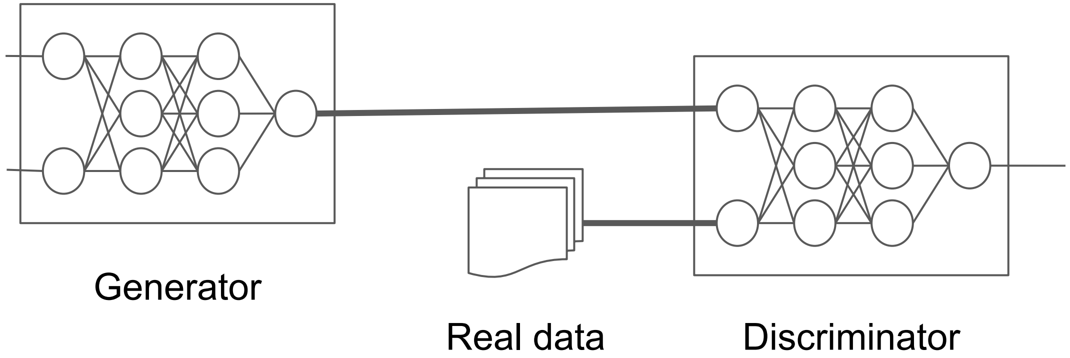


Recap …

Training

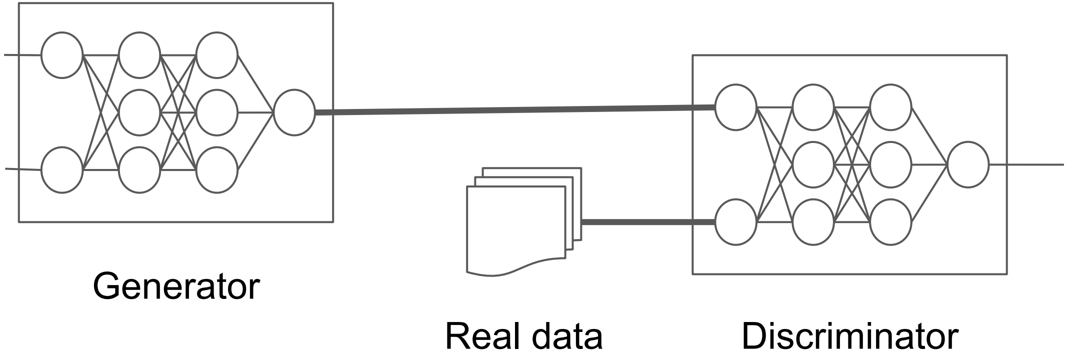
Recap …

Training



Recap …

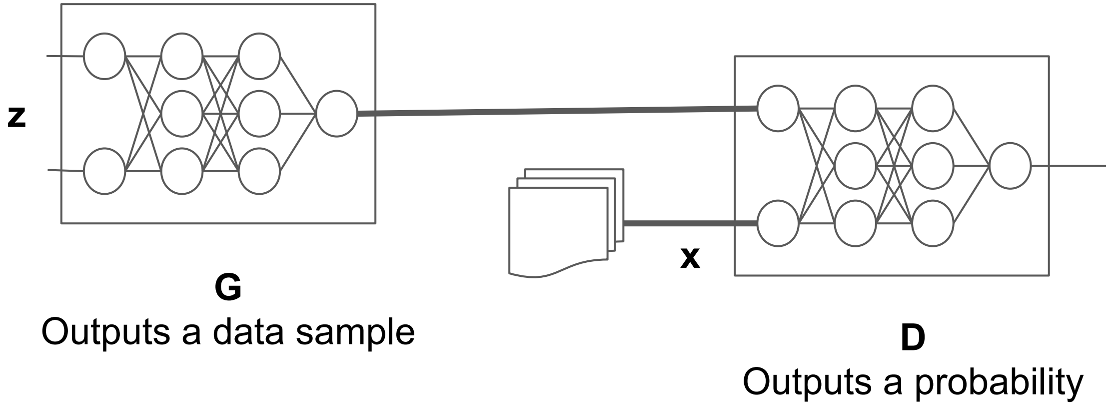
Training



$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Recap … Training

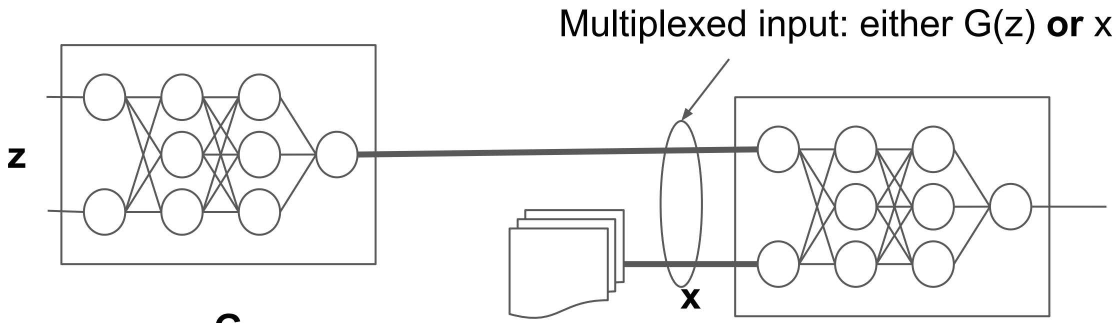


$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Recap …

Training



G
Outputs a data sample

D
Outputs a probability

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Recap …

Training

Use SGD-like algorithm of choice on two minibatches simultaneously:

A minibatch of **training examples**

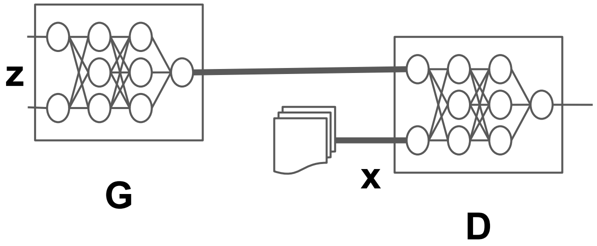
A minibatch of **generated samples**

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

Training intuition

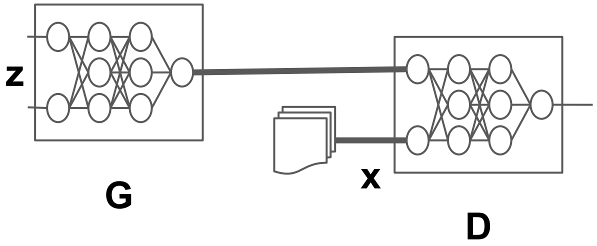


$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

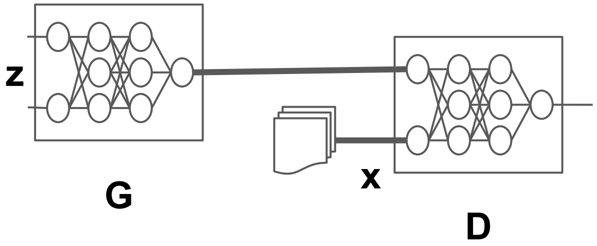
Inputs		Prediction	Consequence	
G(z)	x	D	J(D)	J(G)



$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

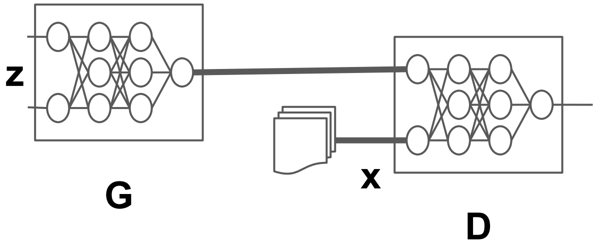


Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓				

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

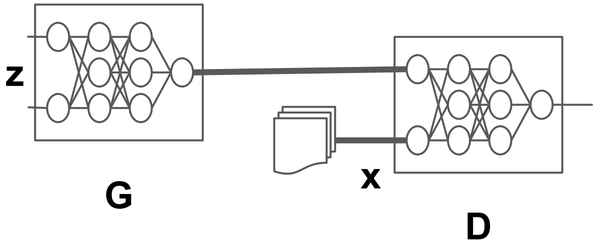


Inputs		Prediction	Consequence	
G(z)	x	D	J(D)	J(G)
✓		G		

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition



Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G		

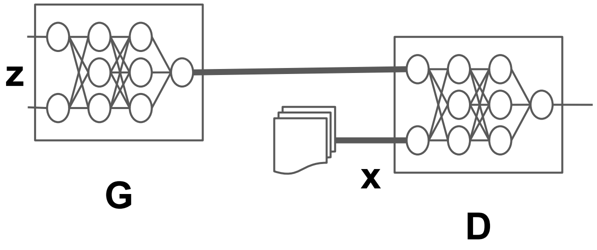
$$J^{(D)} = \boxed{-\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x})} - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

0

Training intuition

Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G		



$$J^{(D)} = $$

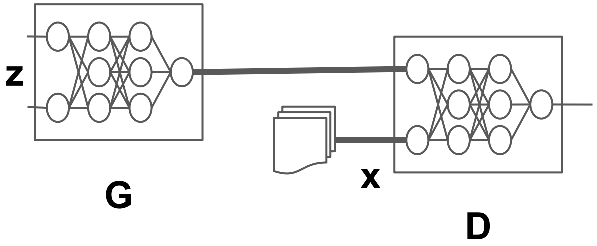
$$J^{(G)} = -J^{(D)}$$

$$-\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

> 0.5

Training intuition

Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G		



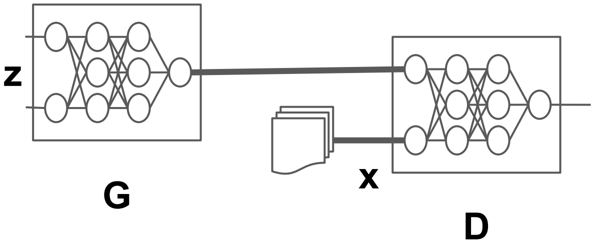
$$J^{(D)} =$$

$$J^{(G)} = -J^{(D)}$$

$$-\frac{1}{2}\mathbb{E}_{z} \log \left(1 - \underbrace{\overbrace{D\left(G(z)\right)}^{> 0.5}}_{< 0.5}\right)$$

Training intuition

Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G		

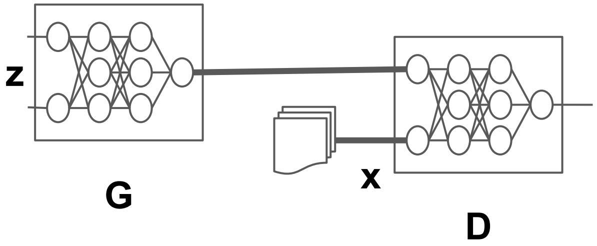


$$J^{(D)} = -\frac{1}{2}\mathbb{E}_z \log \underbrace{\left(1 - \overset{>0.5}{D\left(G(z)\right)}\right)}_{<0.5}$$

log(<0.5) **<** log(>0.5)

$$J^{(G)} = -J^{(D)}$$

Training intuition

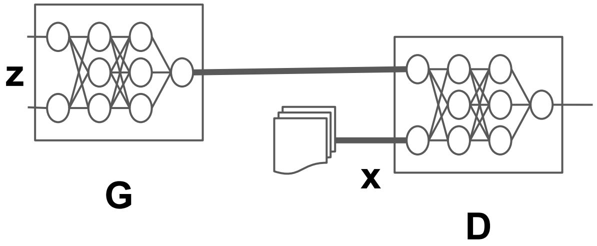


Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G	**>**	

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition



Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G	>	
✓		x		>
	✓	G	>	
	✓	x		>

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log \left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

Inputs		Prediction	Consequence	
G(z)	x	D	J(D)	J(G)
✓		G	>	
✓		x		>
	✓	G	>	
	✓	x		>

Conclusion:
We (ideally) consider the generator trained when

p(D(G(z)) = p(D(x)) = 0.5

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

Training intuition

Conclusion:
We (ideally) consider the generator trained when

p(D(G(z)) = p(D(x)) = 0.5

(Nash equilibrium)

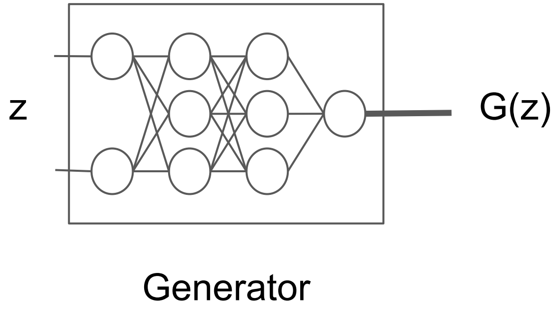
Inputs		Prediction	Consequence	
G(z)	**x**	**D**	**J(D)**	**J(G)**
✓		G	>	
✓		x		>
	✓	G	>	
	✓	x		>

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

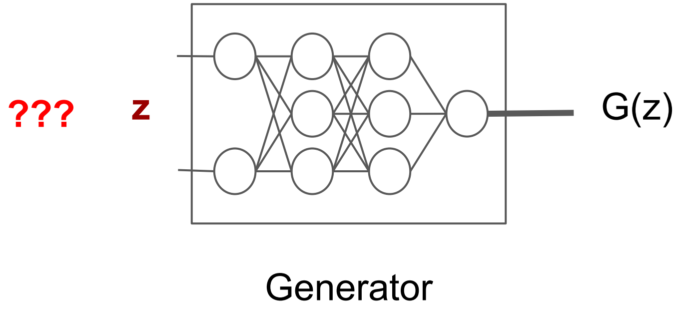
$$J^{(G)} = -J^{(D)}$$

Controlling the output [Mirza et. al]

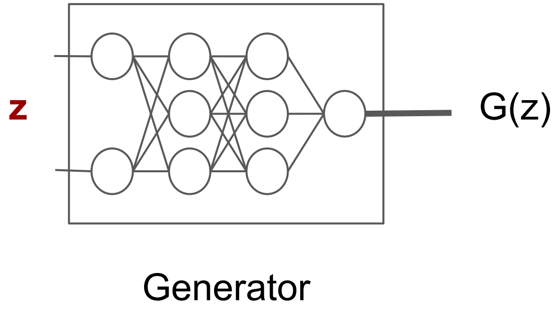
Controlling the output



Controlling the output

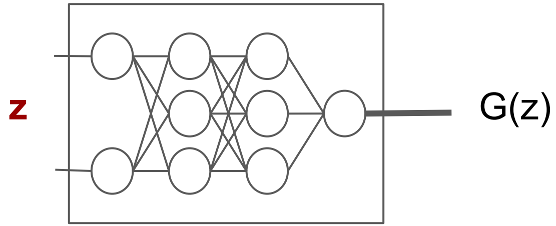


Controlling the output



In an uncontrolled environment, z is sampled uniformly from a normal distribution. (latent space/noisy)

Controlling the output

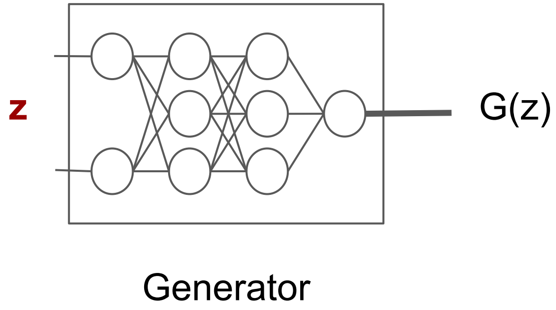


Generator

In an uncontrolled environment, z is sampled uniformly from a normal distribution. (latent space/noisy)

Controlling the output of G means controlling some aspects of its input, z.

Controlling the output

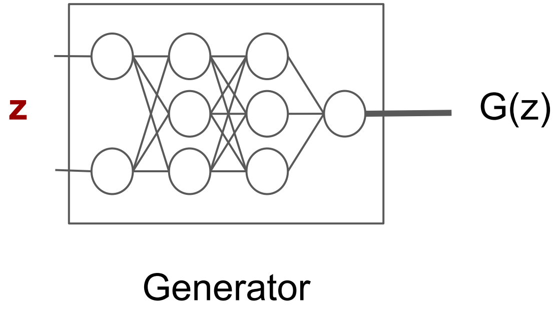


In an uncontrolled environment, z is sampled uniformly from a normal distribution. (latent space/noisy)

Controlling the output of G means controlling some aspects of its input, z.

We do this by encoding features of the training data into z along with its normal sampled values.

Controlling the output



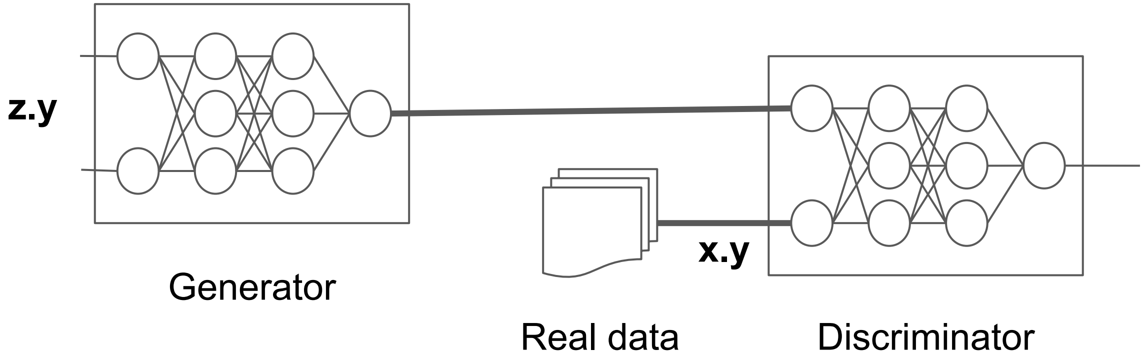
In an uncontrolled environment, z is sampled uniformly from a normal distribution. (latent space/noisy)

Controlling the output of G means controlling some aspects of its input, z.

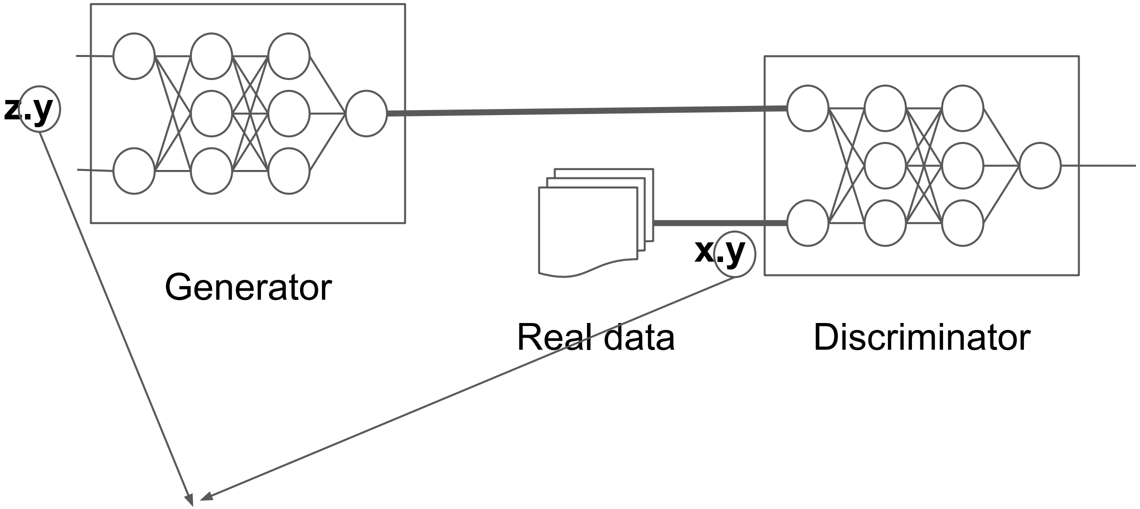
We do this by encoding features of the training data into z along with its normal sampled values.

In a fair manner!

Controlling the output



Controlling the output



Fair = the extra-info is fed, at training time, to both
the generator and the real-sampled data

Code time

Sources

[1] “Unsupervised Image-to-Image Translation Networks” -- Ming-Yu et al.
https://arxiv.org/pdf/1703.00848.pdf

[2] “Artificial intelligence can say yes to the dress” -- Quartz article
https://qz.com/1090267/artificial-intelligence-can-now-show-you-how-those-pants-will-fit/

[3] “Semantic Image Inpainting with Deep Generative Models” -- Yeh et al.
https://arxiv.org/pdf/1607.07539.pdf

[4] “Generative Adversarial Text to Image Synthesis” -- Reed et al.
https://arxiv.org/pdf/1605.05396.pdf

Sources (2)

[5] “AdversarialNetsPapers” -- JiChao Zhang
https://github.com/zhangqianhui/AdversarialNetsPapers

[6] “Conditional Generative Adversarial Nets” -- Mirza et al.
https://arxiv.org/pdf/1411.1784

[7] “Tensorflow Models Github: TfGAN Jupyter Notebook” -- Joel Shor
https://github.com/tensorflow/models/blob/master/research/gan/tutorial.ipynb