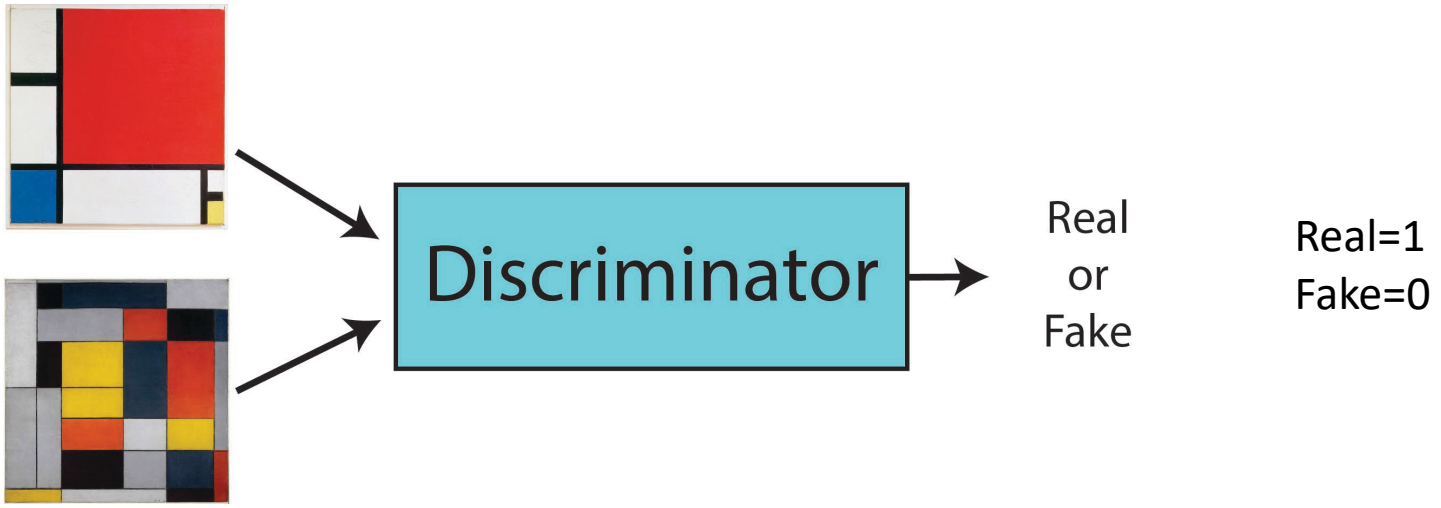
Generative adversarial networks

Lecture 9a

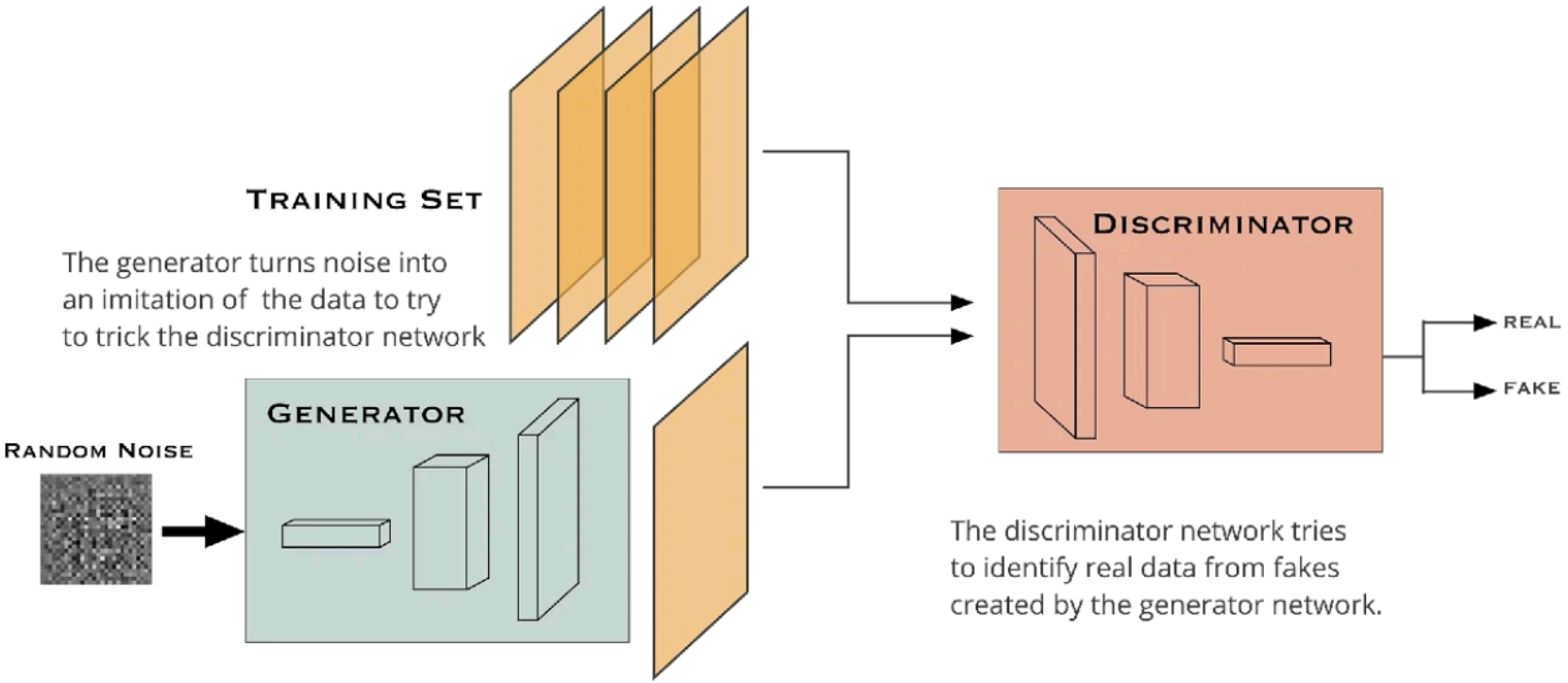
Fake images

Discriminator

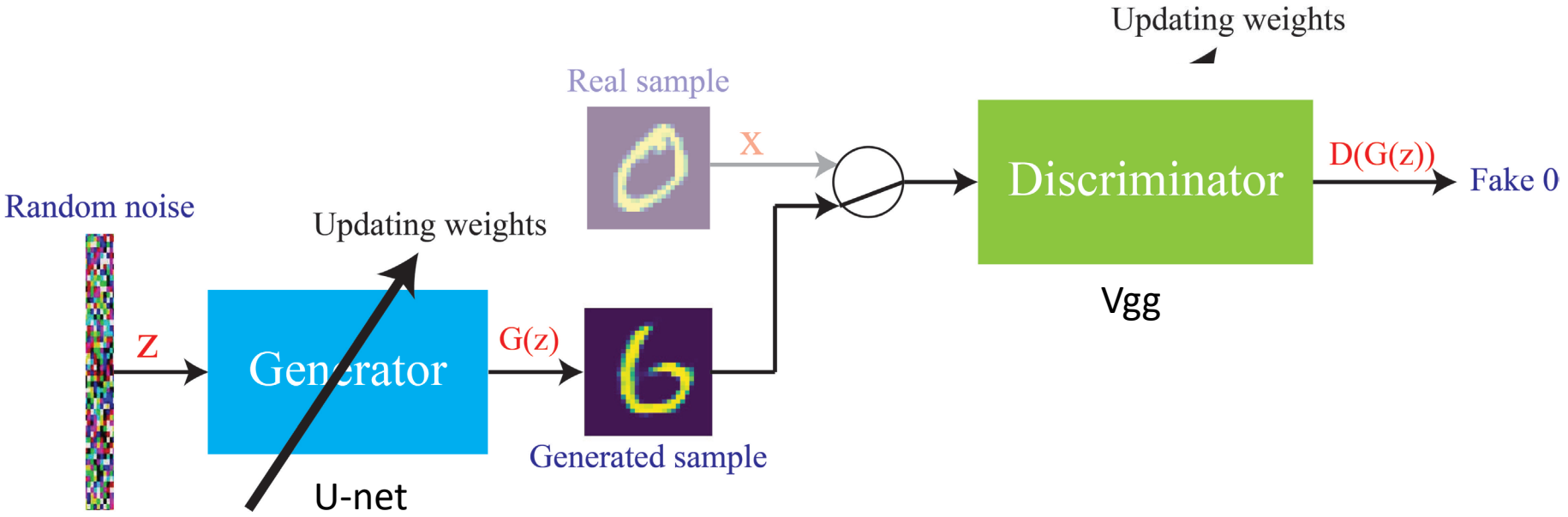
Mondrian paintings



Generative Adversarila Network (GAN)



Generative Adversarial Networks (GAN)



Cost function:

$$C = \sum_{x}\left[\log\left(D\left(x\right)\right)\right] + \sum_{z}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right]$$

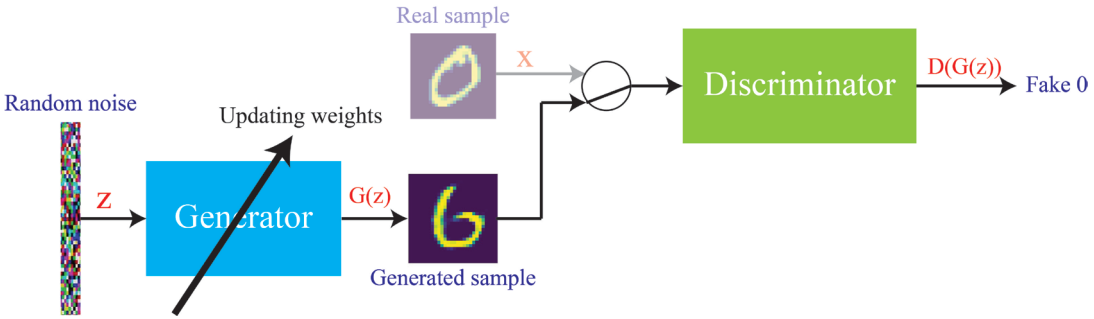
The generator tries to minimize the cost function
The discriminator tries to maximize the cost function

- x is the real data
- z is the random noise
- D(x) is the discriminator output
- G(z) is the generated data

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

Extreme behaviour (D= 0 or 1)

Real input	D=1	C=logD=0	DN no change	GN no change
Real input	D=0	C=log(0)=-∞	DN changes	GN no change
Fake input	D=1	C=log(1-D)=−∞	DN changes	GN no change
Fake input	D=0	C=log(1-D)=0	DN no change	GN changes



$$C = \sum_{x}\left[\log\left(D\left(x\right)\right)\right] + \sum_{z}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right]$$

GAN: generating digits

Generator

Layer	Size out	Activation	Kernel
Input	100		
Fully connected	100→12544	LeakyReLU	
Reshape	7x7x256		
Convolution	7x7x128	LeakyReLU	5x5
Up-sampling	14x14x128		
Convolution	14x14x64	LeakyReLU	5x5
Up-sampling	28x28x64		
Convolution	28x28x1	tanh	5x5

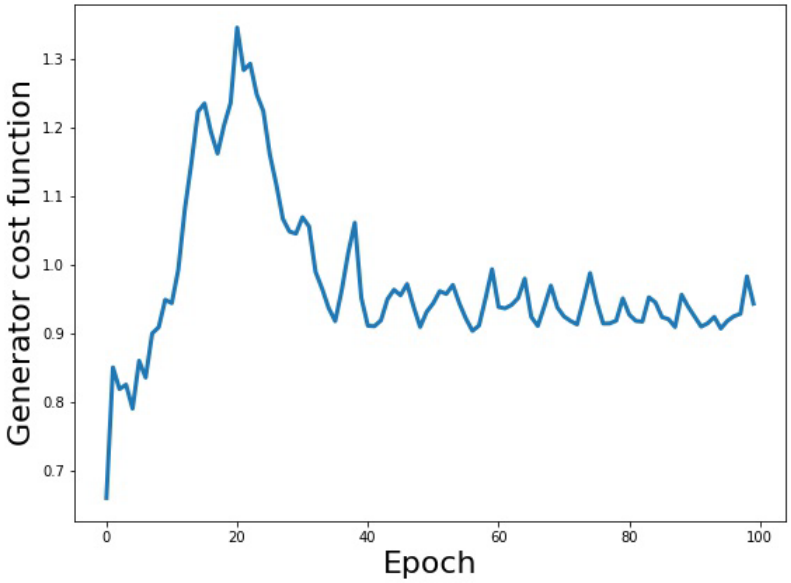
Discriminator

Layer	Size out	Activation	Kernel
Input	28x28x1		
Convolution	28x28x64	LeakyReLU	5x5
Down-sampling	14x14x64		
Dropout			
Convolution	14x14x128	LeakyReLU	5x5
Down-sampling	7x7x128		
Dropout			
Fully connected	6272→1		

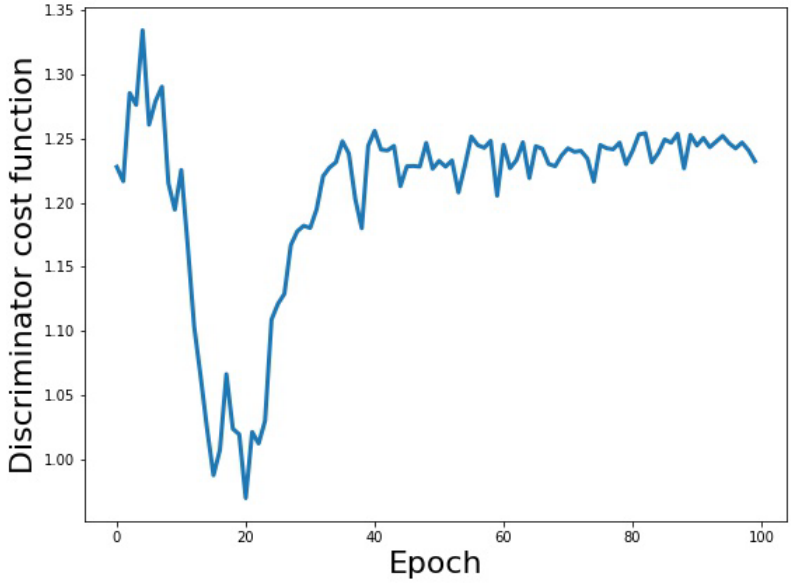
- Mnist digits as real samples
  - 60000 real samples
- Generator
  - Adam optimizer
  - Learning rate=0.0001
- Generator input:
  - Random normal between 0 and 1
- Discriminator
  - Adam optimizer
  - Learning rate=0.0001
- 100 epochs

GAN: generating digits

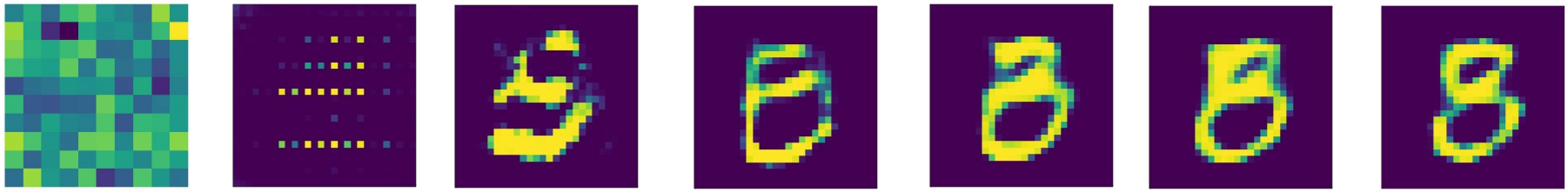
Generator



Discriminator



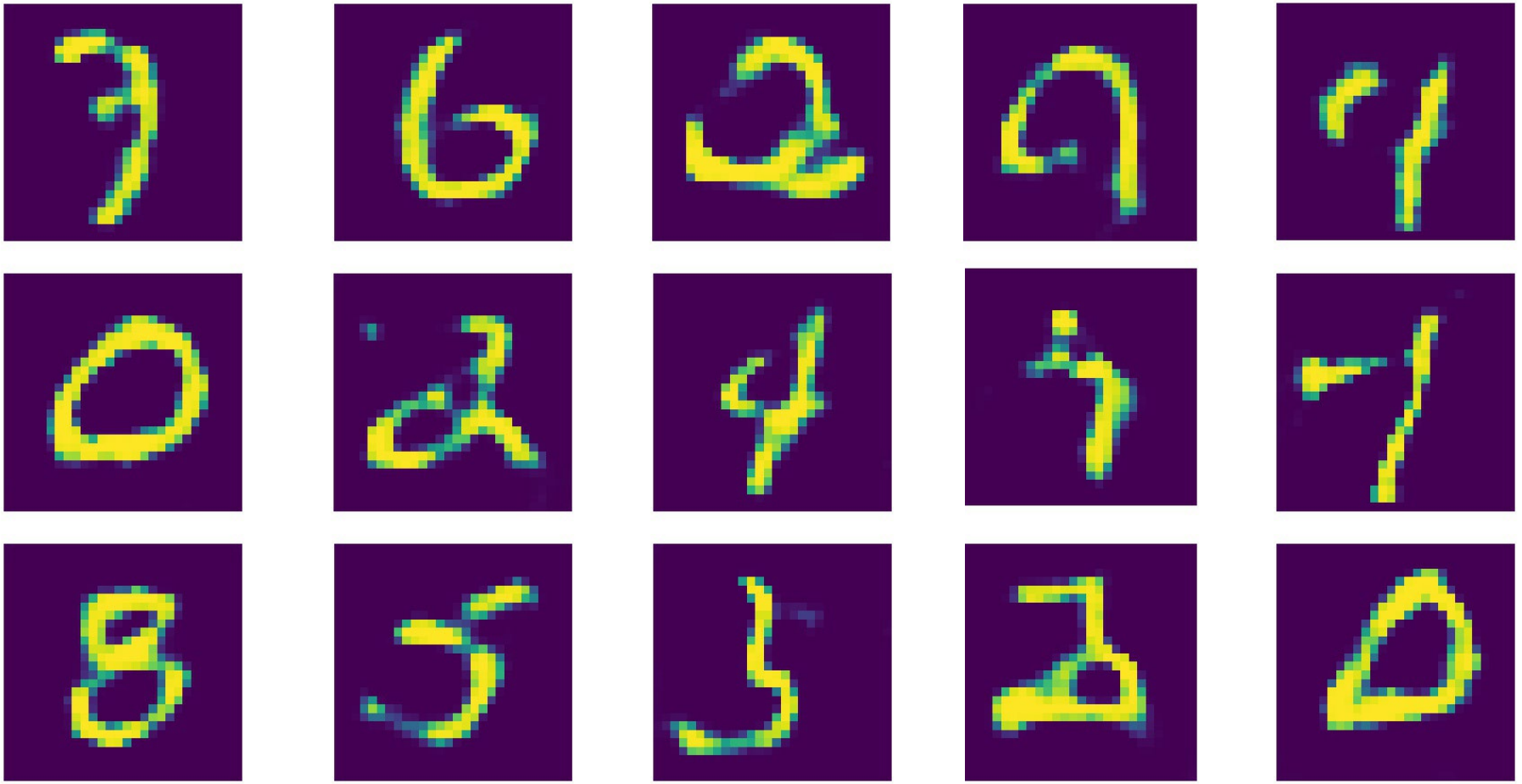
Generator input | Epoch 1 | Epoch 10 | Epoch 30 | Epoch 60 | Epoch 80 | Epoch 100



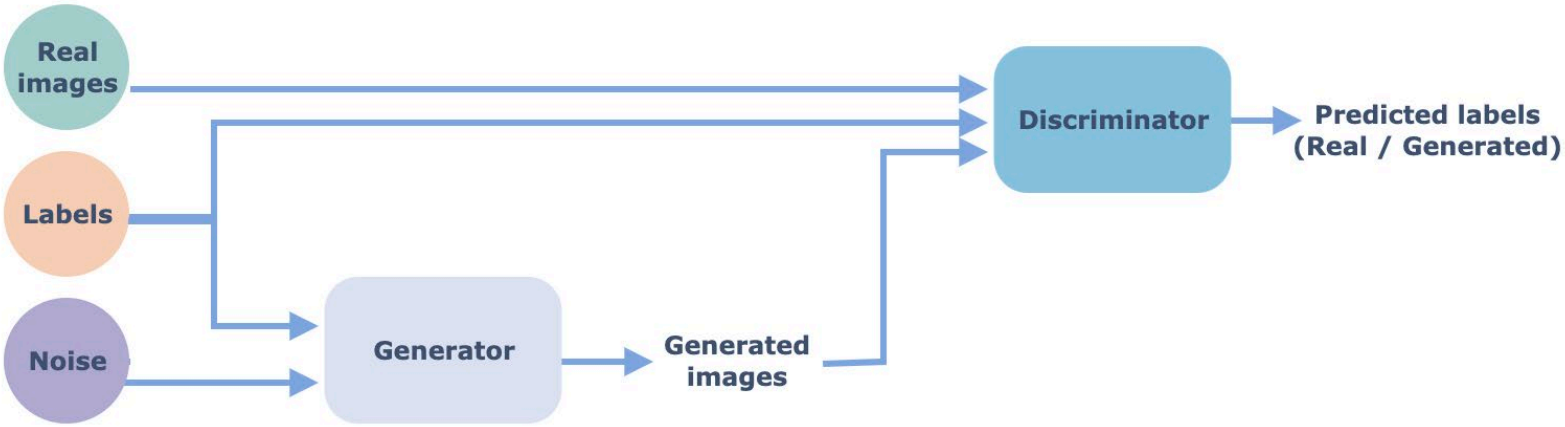
GAN: generating digits

Examples of generated digits

There is no control over which digit is going to be generated

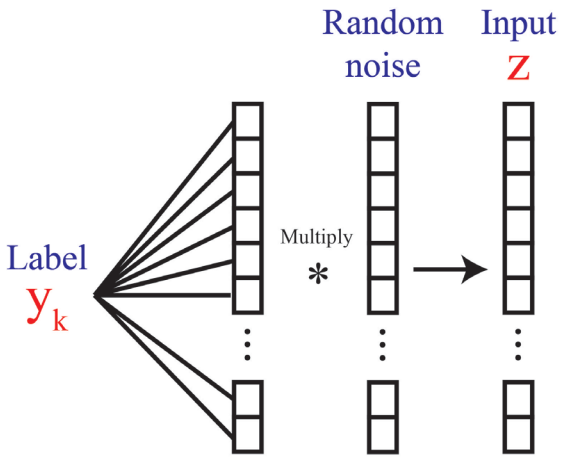


Conditional GAN



Conditional GAN

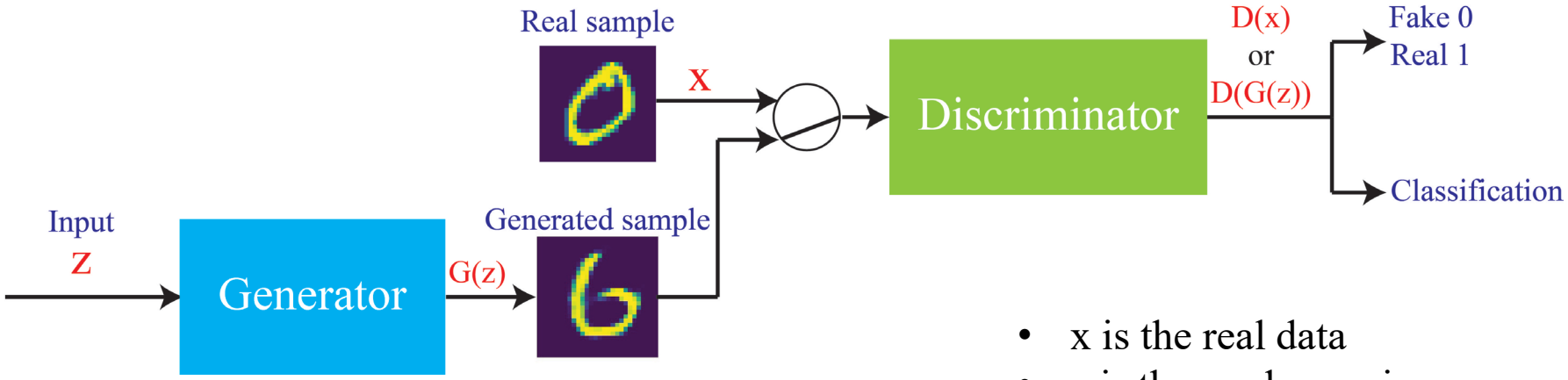
Auxiliary classifier GAN: AC-GAN



Cost function:

$$C_{\text{discrimination}} = \sum_x \left[\log\left(D\left(x\right)\right)\right] + \sum_z \left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right]$$

$$C_{\text{classification}} = \sum_x \left[\sum_k \left(y_k^{(x)} \log\left(D\left(x\right)\right)\right)\right] + \sum_z \left[\sum_k \left(y_k^{(z)} \log\left(D\left(G\left(z\right)\right)\right)\right)\right]$$



- Discriminator maximizes $C_{\text{classification}} + C_{\text{discrimination}}$
- Generator maximizes $C_{\text{classification}} - C_{\text{discrimination}}$

- x is the real data
- z is the random noise
- G(z) is the generated data
- D(x) is the discriminator output
- $y_k^{(x)}$ is the label of sampel x

Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

AC-GAN: generating digits

Generator

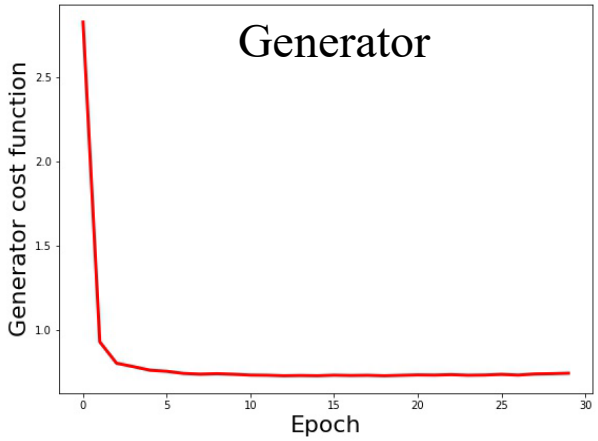
Layer	Size out	Activation	Kernel
Input	100		
	1→100		
Multiplication of inputs	100x2→100		
Fully connected	100→6272	ReLU	
Reshape	7x7x128		
Up-sampling	14x14x128		
Convolution	14x14x128	ReLU	3x3
Up-sampling	28x28x128		
Convolution	28x28x64	ReLU	3x3
Convolution	28x28x1	tanh	3x3

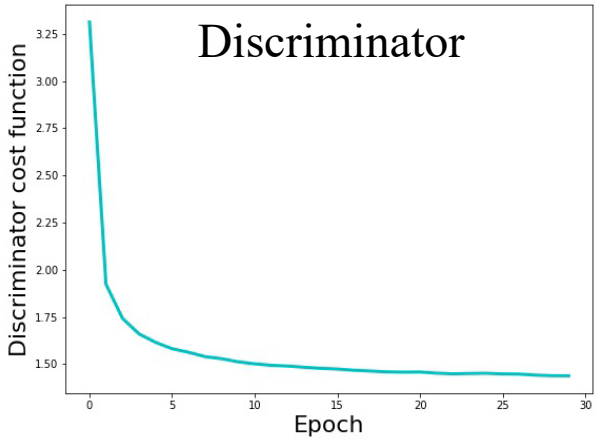
Discriminator

Layer	Size out	Activation	Kernel
Input	28x28x1		
Convolution	28x28x16	LeakyReLU	3x3
Down-sampling	14x14x16		
Dropout			
Convolution	14x14x32	LeakyReLU	3x3
Down-sampling	7x7x32		
Zero-padding	8x8x32		
Dropout			
Convolution	8x8x64	LeakyReLU	3x3
Down-sampling	4x4x64		
Dropout			
Convolution	4x4x128	LeakyReLU	3x3
Dropout			
Output	2048→1	Sigmoid	
Fully connected	2048→10	Softmax	

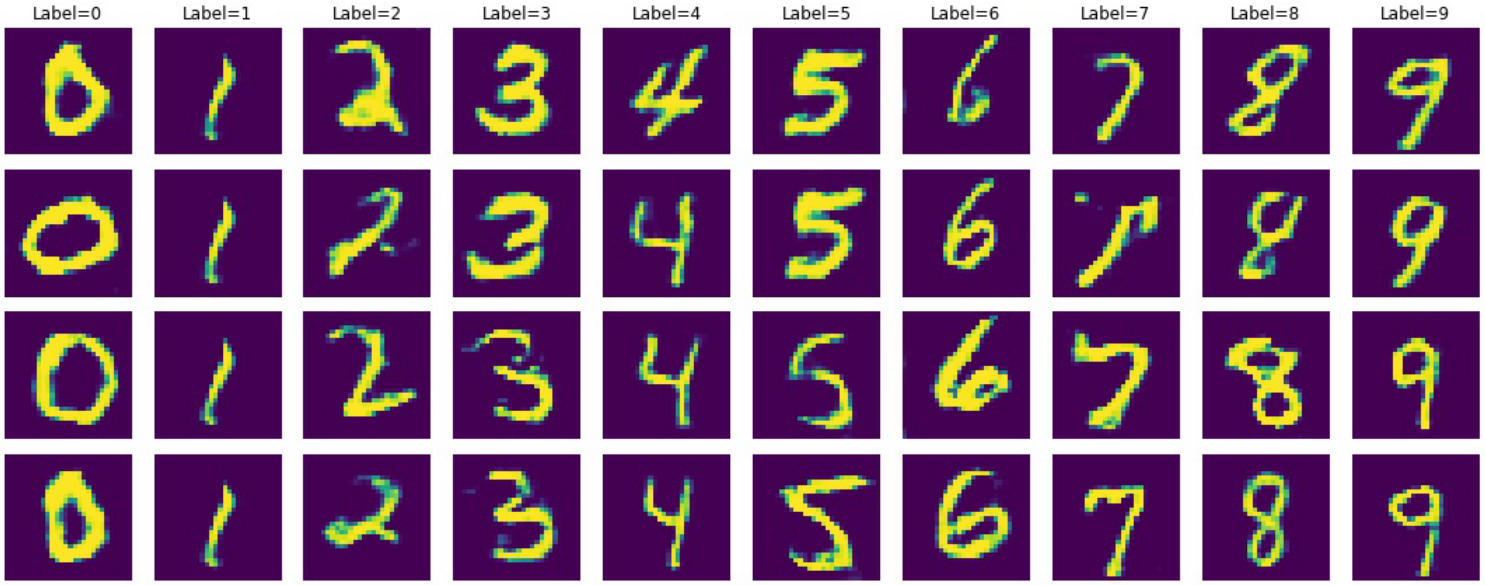
- Mnist digits as real samples
  - 60000 real samples
- Generator
  - Adam optimizer
  - Learning rate=0.0002
- Generator inputs:
  - Random normal between 0 and 1
  - Label of the digit
- Discriminator
  - Adam optimizer
  - Learning rate=0.0002
- Discriminator outputs:
  - Determining if the input is real or fake
  - Classifying the input image
- 30 epochs

AC-GAN: generating digits



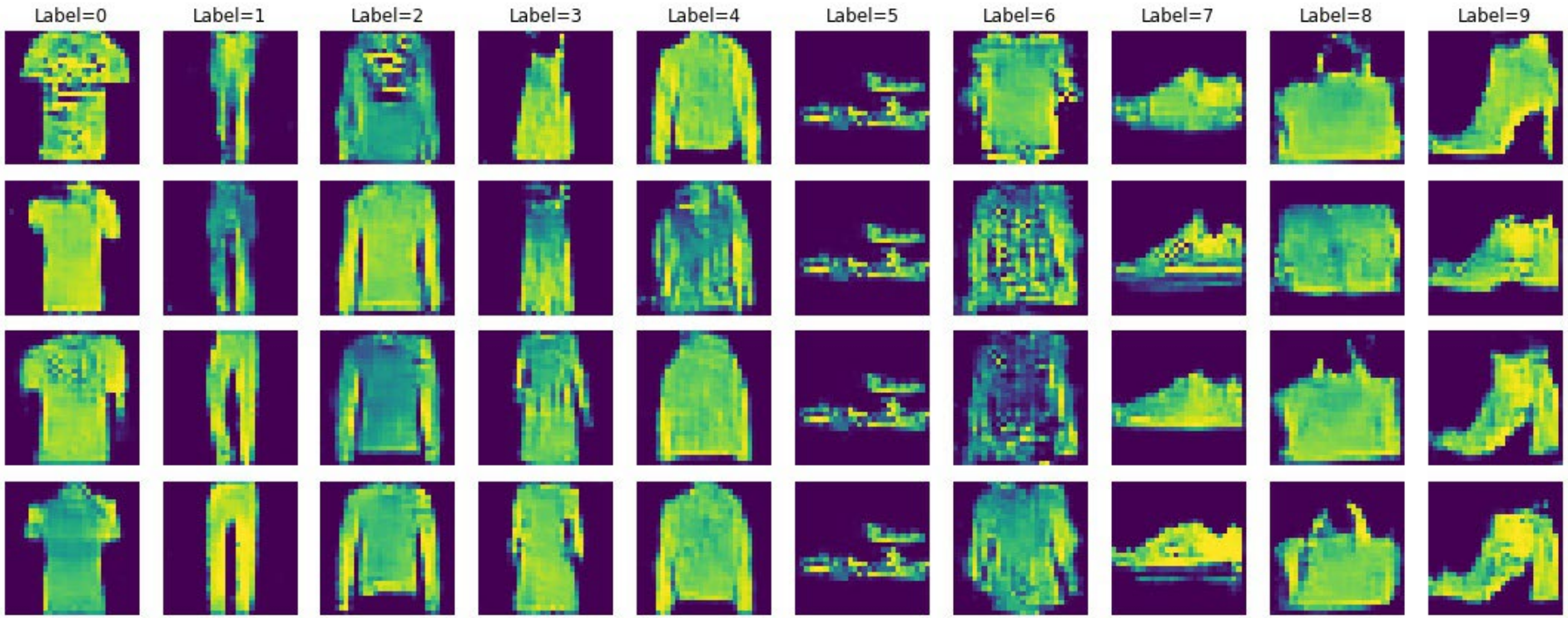


Examples of generated digits

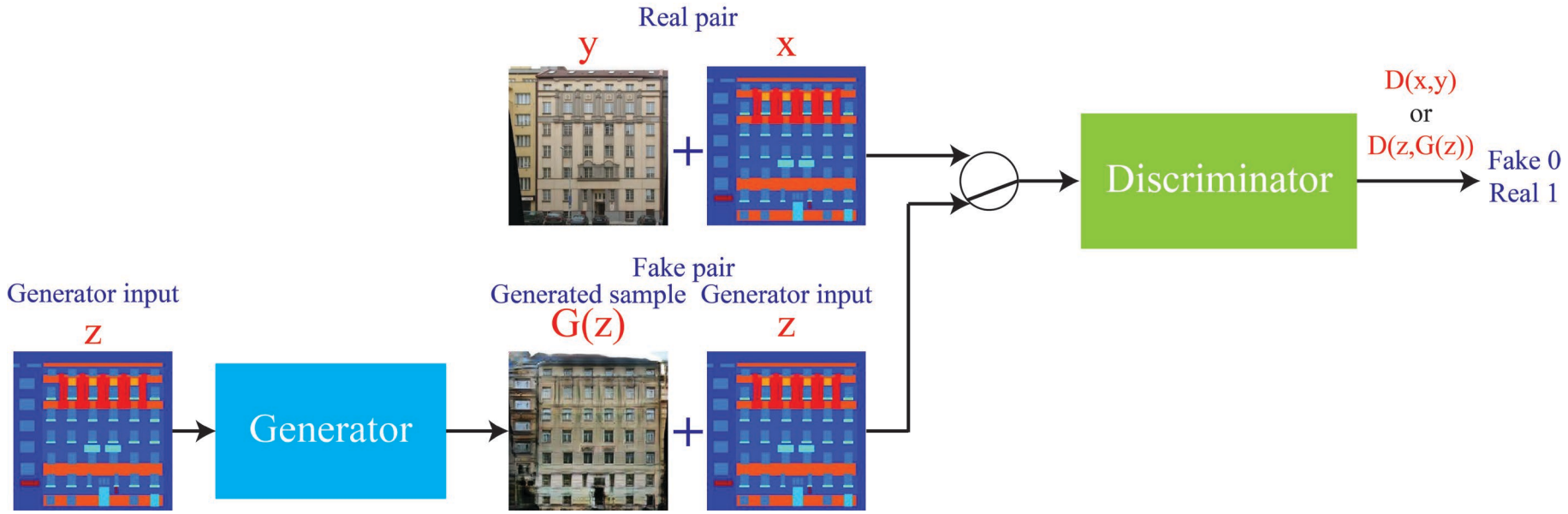


AC-GAN: generating images like mnist-fashion

Examples of generated garments



GAN: Image-to-Image translation



Cost function:

$$C = \sum_{x,y}\left[\log\left(D\left(x,y\right)\right)\right] + \sum_{z}\left[\log\left(1 - D\left(z, G\left(z\right)\right)\right)\right]$$

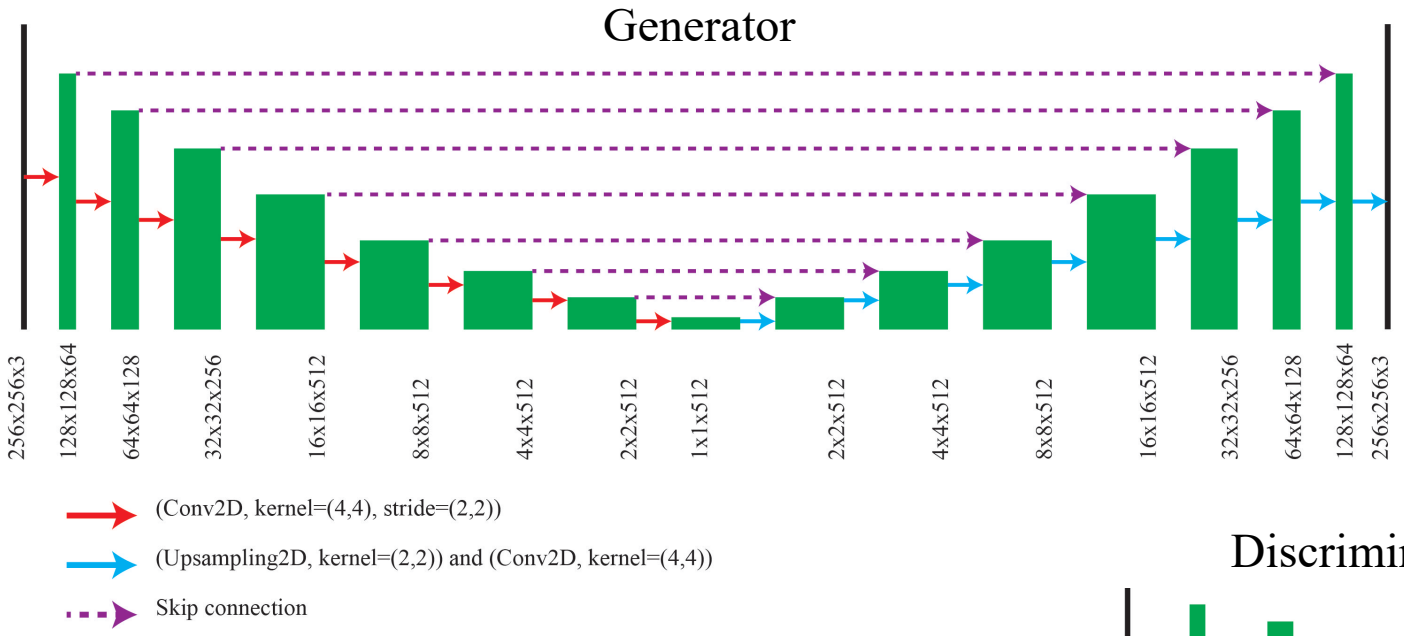
$$C_{L1} = \sum_{z,y}\left\| y - G\left(z\right)\right\|_1$$

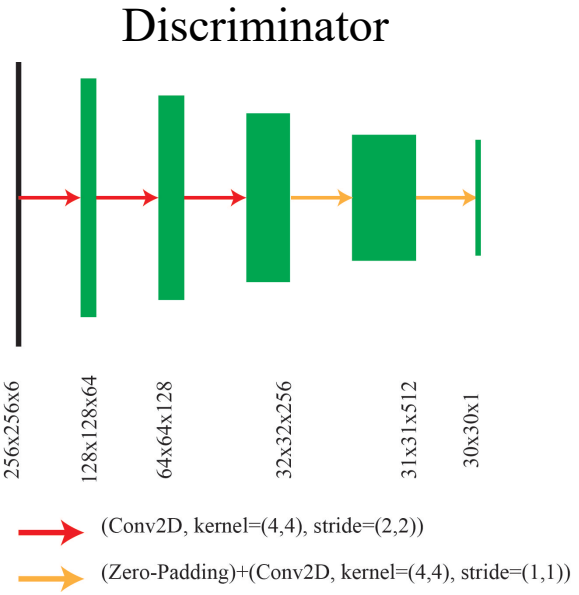
- Discriminator maximizes C
- Generator minimizes C + C_{L1}

- x and z are sketches of buildings
- y is the target image
- G(z) is the generated data
- D(x,y) is the discriminator output

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

GAN: Image-to-Image translation

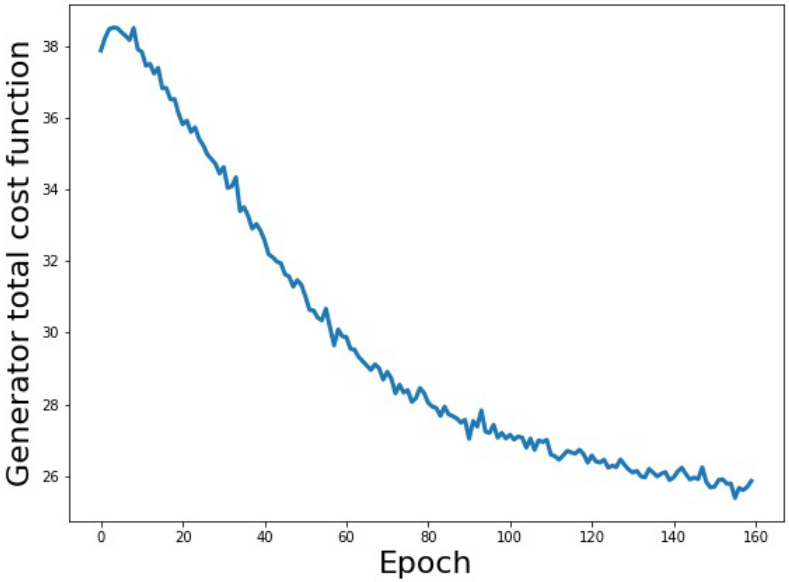




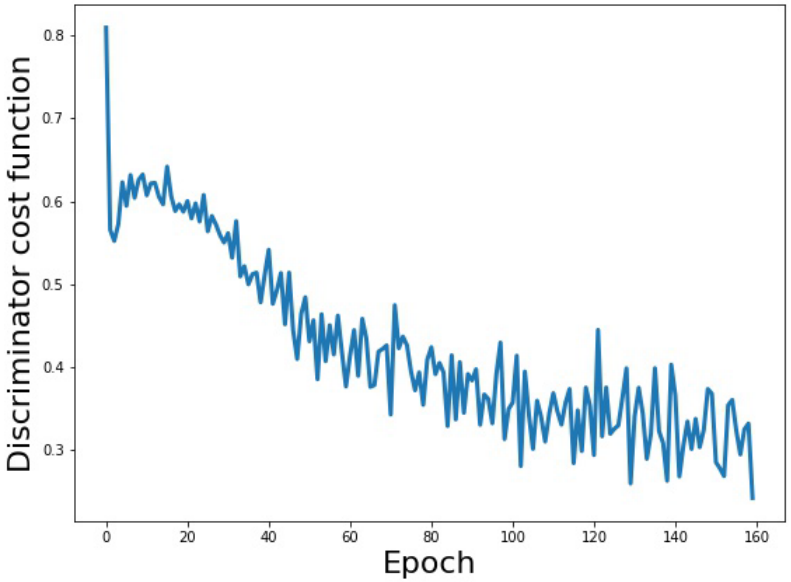
- Generator
  - Adam optimizer
  - Learning rate=0.0002
- Discriminator
  - Adam optimizer
  - Learning rate=0.0002
- 160 epochs

GAN: Image-to-Image translation

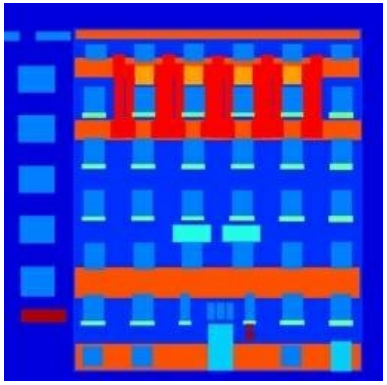
Generator



Discriminator



Generator input



Target

GAN

GAN: Image-to-Image translation