# Lecture 14: GAN: Loss Functions and De-convolution

Madhavan Mukund and Pranabendu Misra

Advanced Machine Learning 2021

- Another class of *Generative* models.
- **The generation step is at the heart of this model.**

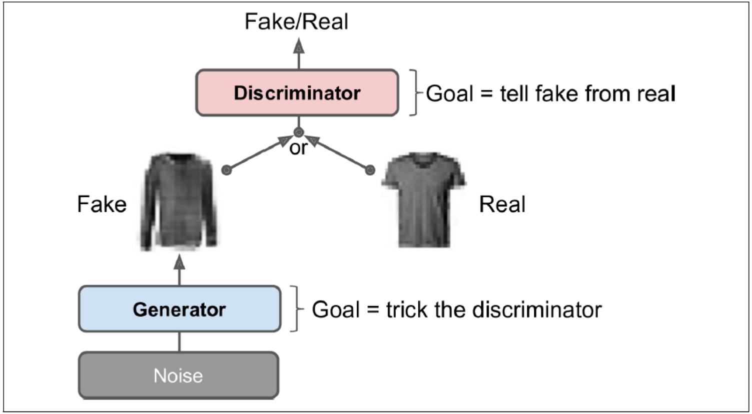



*Figure 17-15. A generative adversarial network*

# Generative Adversarial Network

- **Generator**: A Neural Network which given some *random noise* it generates a *fake datapoint*.
- **Discriminator**: A Neural Network that given an image determines if it is real or fake.

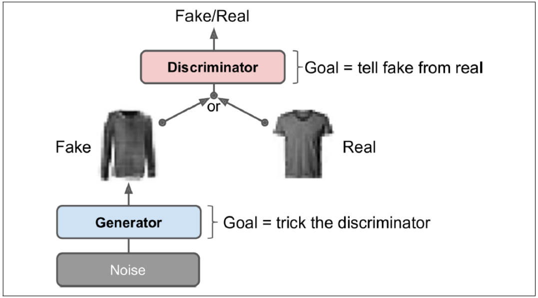



*Figure 17-15. A generative adversarial network*

This is a game-theoretic modeling of the problem.

- Generator and Discriminator are both trying to outwit each other.
- In this process, the Generator learns to produce very good *fake data*.

# Generative Adversarial Network: Loss Functions and Training

- A *Sampler* $S$ samples a vector $z$ in the *latent space* using a normal distribution.
- The Generator $G$ maps $z$ to a data point.

$$\hat{x} = G(z)$$

- The Discriminator $D$ gets $\{\hat{x}, x\}$ where $x$ is a training sample, i.e. a real datapoint.
- $D$ outputs a probability for both $\hat{x}$ and $x$ of them being real (i.e. not generated by $G$)

$$\hat{y} = D(\hat{x})$$

$$y = D(x)$$

- Note that $\hat{y}$ and $y$ are probabilities

# Generative Adversarial Network: Loss Functions and Training

- Discriminator's Objective:

$$\min_{w(D)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

- Generator's Objective:

$$\max_{w(G)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

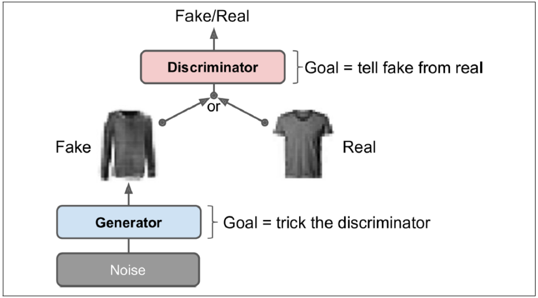



*Figure 17-15. A generative adversarial network*

# Generative Adversarial Network: Loss Functions and Training

- Discriminator's Objective:

$$\min_{w(D)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

- Generator's Objective:

$$\max_{w(G)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

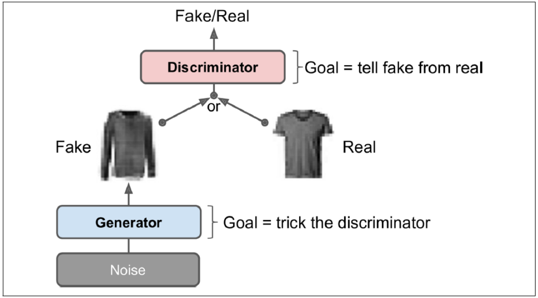



*Figure 17-15. A generative adversarial network*

- This is the cross-entropy loss
- The input $x$ to the Discriminator is labeled 1 with probability $1/2$.
- The predicted label is 1 with probability $D(x)$
- The generator has to *maximize the discriminator's loss*

# Generative Adversarial Network: Loss Functions and Training

- Discriminator's Objective:

$$\min_{w(D)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

- Discriminator's Loss:

$$-\log(D(x)) - \log(1 - D(G(z)))$$

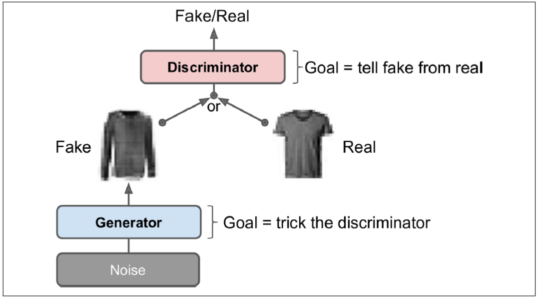


*Figure 17-15. A generative adversarial network*

- We train in rounds
- The Generator produces a batch of fake data $G(z)$ using random noise $z$.
- A batch of real data $x$ and $G(z)$ are both run through the discriminator, and loss $-\log(D(x)) - \log(1 - D(G(z)))$ is computed.
- We then update $D$ via a gradient descent step, keeping $G$ fixed.

- Generator's Objective:

$$\max_{w(G)} \{-\log(D(x)) - \log(1 - D(G(z)))\}$$

- Generator's Loss:

$$\log(1 - D(G(z)))$$

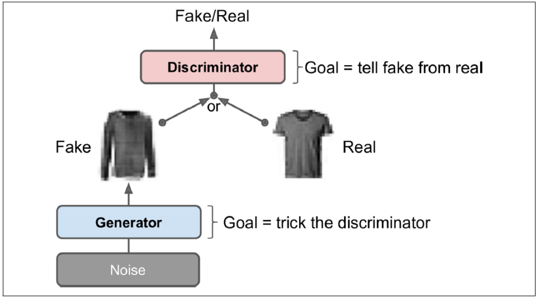
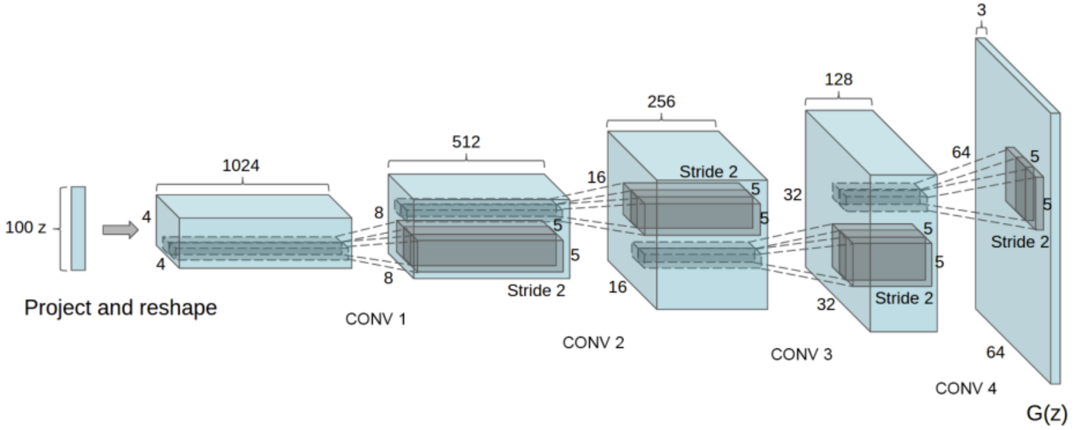


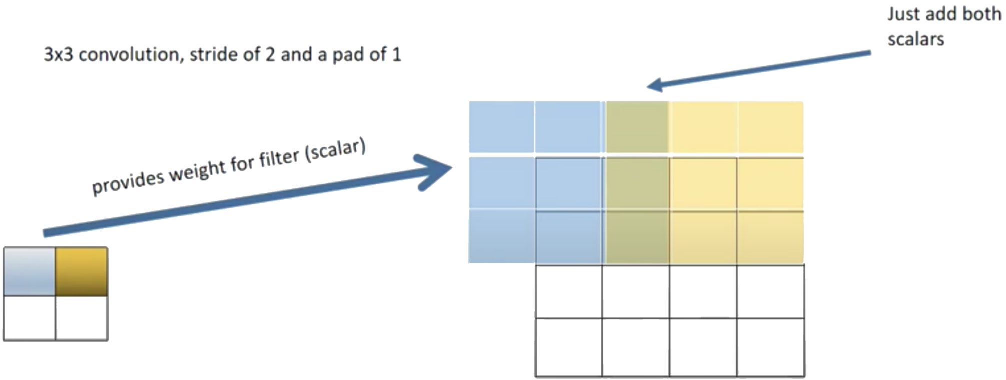

*Figure 17-15. A generative adversarial network*

- Next, the generator then produces a new batch of fake data $G(z')$
- These are processed by the discriminator to get the loss $\log(1 - D(G(z')))$.
- We then update $G$ by a gradient descent step, keeping $D$ fixed.

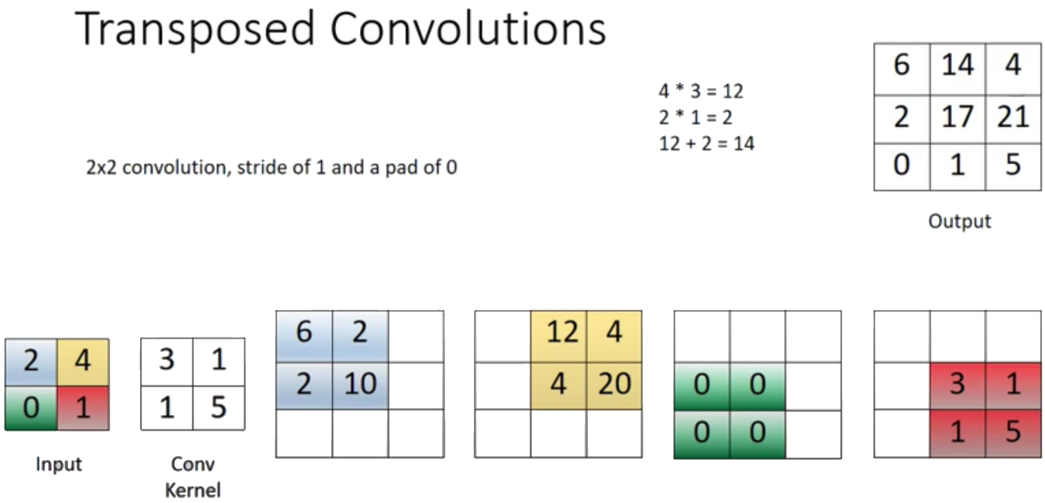
# Deep Convolutional Generator



## Transposed Convolutions



# Deep Convolutional Generator



# References:

These slides are based on:

- http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L4.pdf
- Generative Adversarial Networks, Goodfellow at al, Communications of ACM, November 2020
- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- https://d2l.ai/chapter_generative-adversarial-networks/gan.html
- https://d2l.ai/chapter_computer-vision/transposed-conv.html