Lecture 9: Generative Models

Areej Alasiry

CIS 6217 – Computer Vision for Data Representation

College of Computer Science, King Khalid University

Outline

1. PixelRNN/PixelCNN
2. Generative Adversarial Network (GAN)

Learning Outcomes

1. Explain generative modeling concept and purpose in vision
2. Describe architecture and working of GANs
3. Differentiate **generator** and **discriminator** networks
4. Discuss GAN training and evaluation methods
5. Explore generative models in visual creativity and AI applications

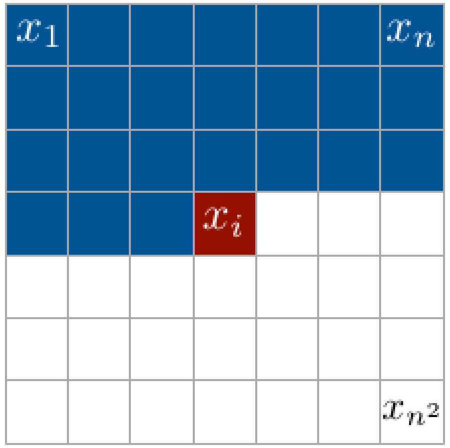
What Are Generative Models?

PixelRNN/PixelCNN

Generative model that generates images pixel by pixel introduced by the researchers at Google DeepMind in 2016

Overview

- Autoregressive generative model that predicts each pixel conditioned on previous ones.
- Learns the joint distribution:
- $P(x) = \prod_{i=1}^{n^2} p(x_i | x_1, \dots, x_{i-1})$
- Generates images pixel-by-pixel in raster-scan order (left → right, top → bottom).



Context

Ingham, F. (2019, May 1). *Day 4: Pixel Recurrent Neural Networks*. Medium. https://medium.com/a-paper-a-day-will-have-you-screaming-hurray/day-4-pixel-recurrent-neural-networks-1b3201d8932d

Architecture

- **Input:** Image pixels (RGB) encoded as discrete values → embeddings.
- **Masked Convolution:**
  - Ensures each pixel only sees *past* pixels (above and left).
  - Type A mask (first layer), Type B (subsequent layers).
- **Recurrent Layers:**
  - **Row LSTM:** scans each row sequentially.
  - **Diagonal BiLSTM:** captures dependencies across rows and columns in parallel.
- **Output:**
  - Predicts a **probability distribution (softmax)** for each pixel value.
  - Autoregressive over color channels (R → G → B).

PixelCNN

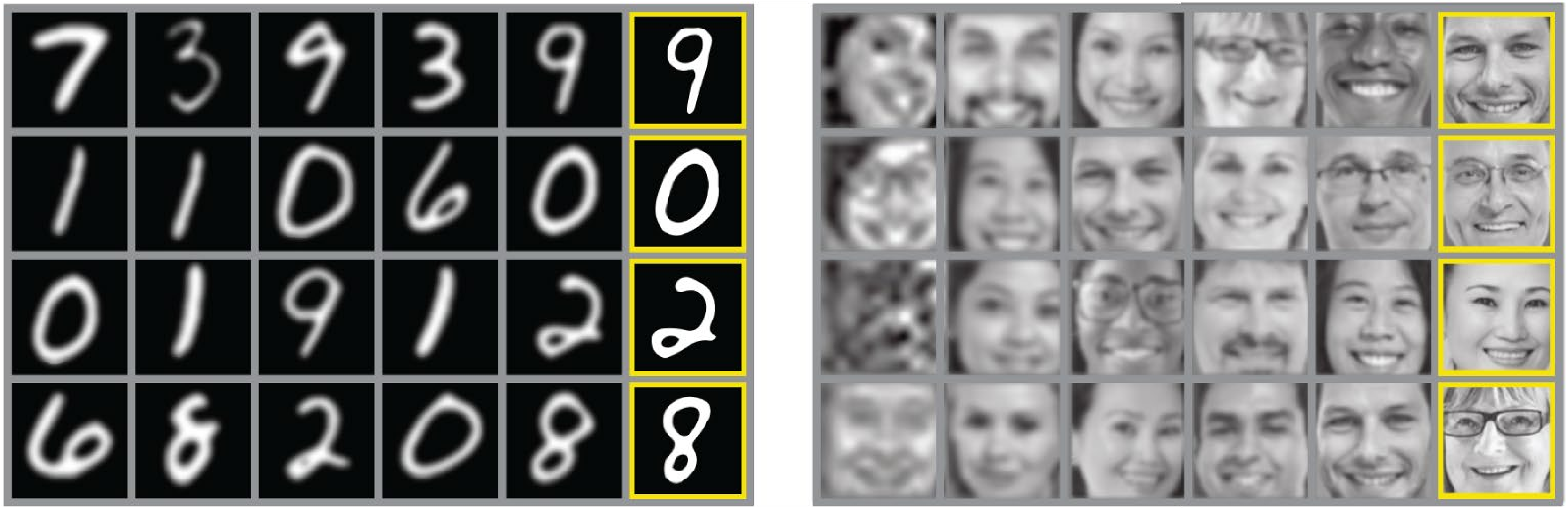
- **Autoregressive generative models** that predict images **pixel-by-pixel**.

Feature	PixelRNN	PixelCNN
Architecture	RNN	CNN
Generation Process	Sequential	More parallel using masked convolutions
Training speed	Slow	Fast
Dependency Modelling	Captures long-range	Capture local context

GAN

Generative Adversarial Networks

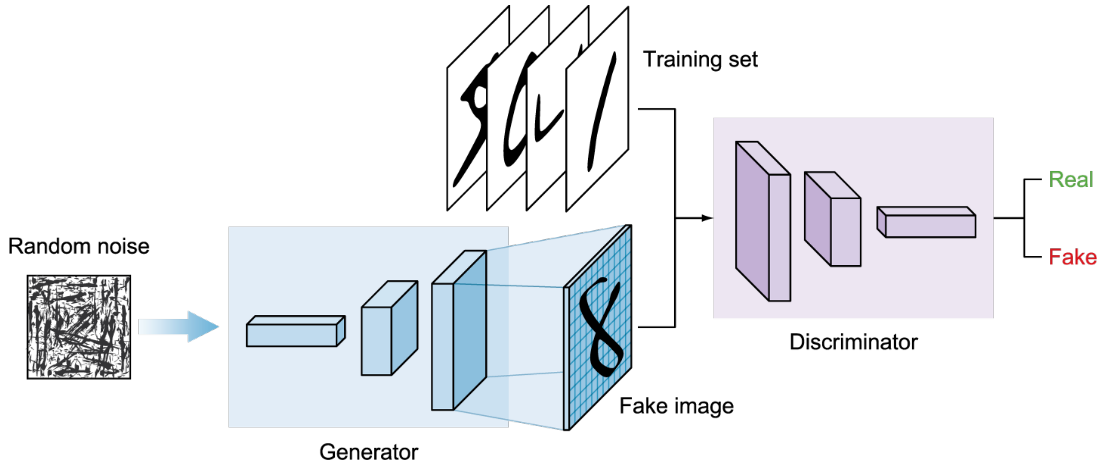
Generative Adversarial Networks



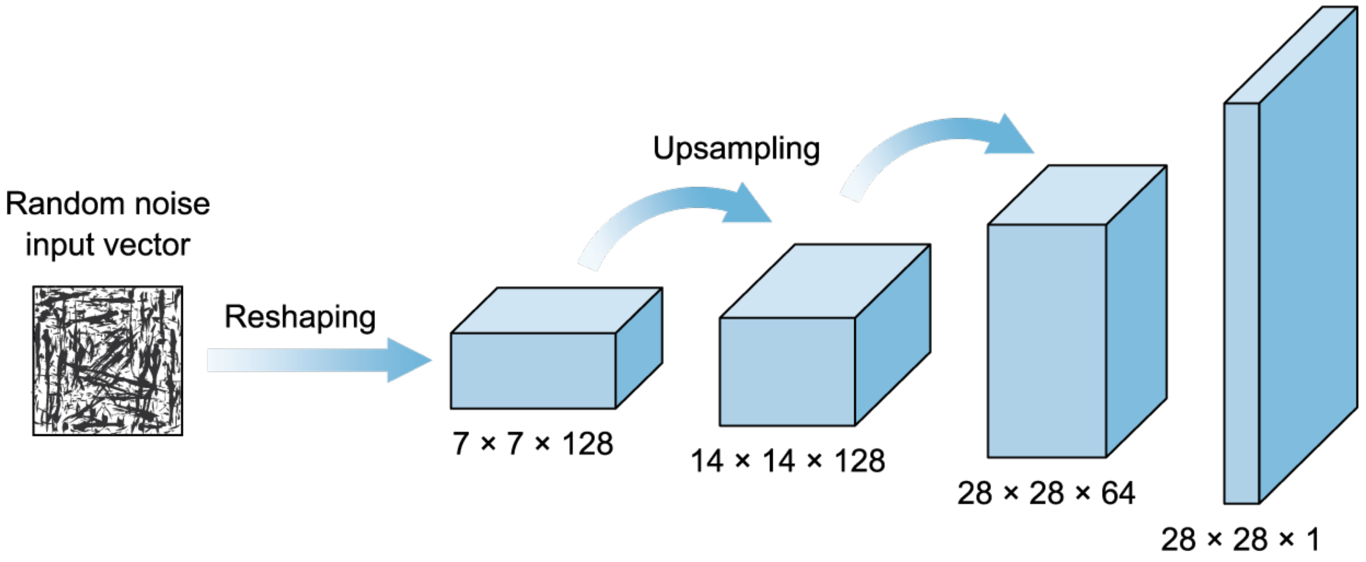
Deep Learning for Vision Systems by Mohamed Elgendy (2020)

GAN Architecture

- Two main Neural Networks:
- Generator
  - Generates images from the features learned from the training dataset
- Discriminator
  - Predict whether the image is real or fake



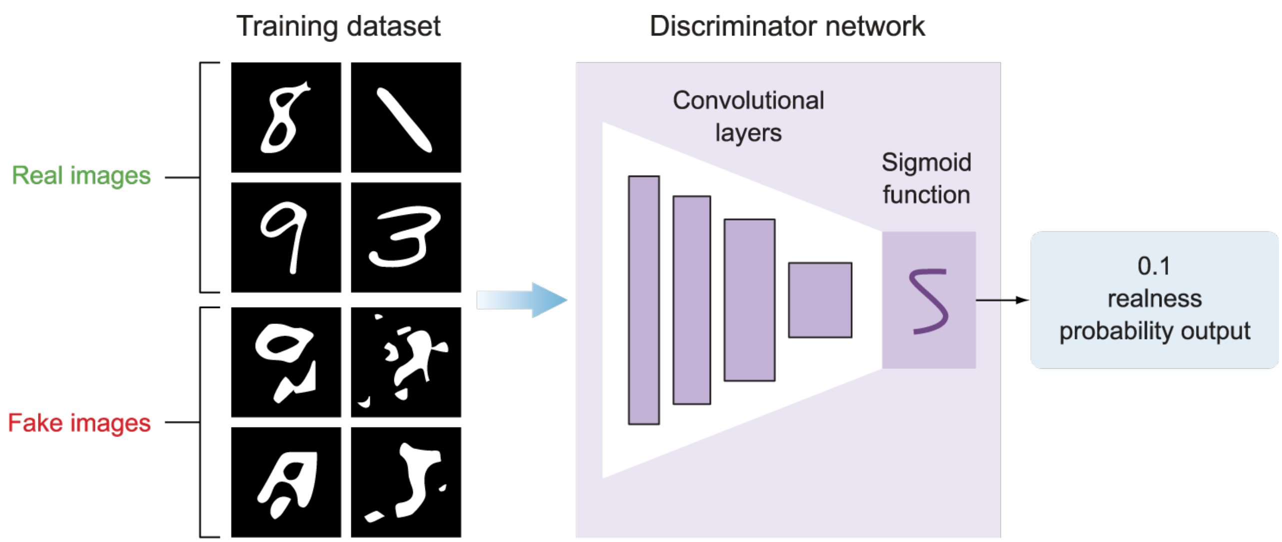
Deep Learning for Vision Systems by Mohamed Elgendy (2020)



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

The Generator Model

Uses random data and tries to mimic the training dataset to generate fake images



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

The Discriminator Model

Predict whether an image is real or fake. Consisting stacked convolution layers followed by a dense output layer with sigmoid activation function

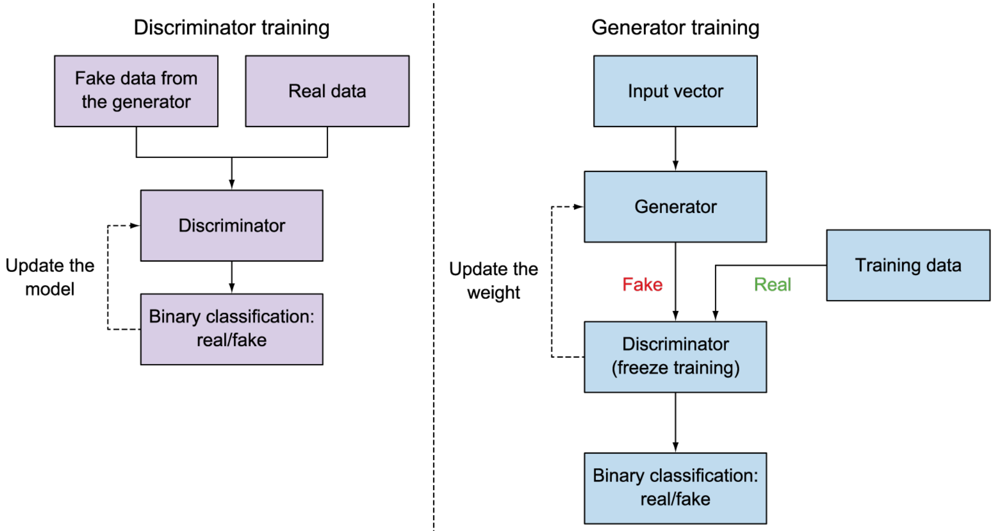
Upsampling

- Traditional CNN uses pooling layer to downsample input image

- Upsampling is used in generative model in order to scale the image dimensions by repeating each row and column of the input pixels.

Training GAN

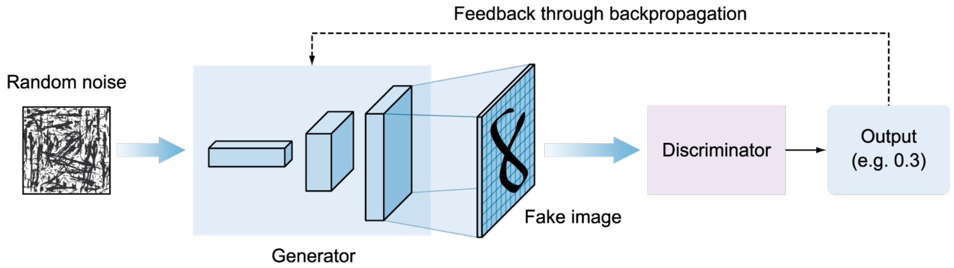
- Train the discriminator.
  - The network is given labeled images coming from the generator (fake) and the training data (real), and it learns to classify between real and fake images with a sigmoid prediction output.
- Train the generator.
  - It needs the discriminator model to tell it whether it did a good job of faking images. So, we create a combined network to train the generator, composed of both discriminator and generator models.



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Training the Generator (Combined)

- When we want to train the generator, we freeze the weights of the discriminator model because the generator and discriminator have different loss functions pulling in different directions.



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Training Epochs

- For each epoch, the two compiled models (discriminator and combined) are trained simultaneously. During the training process, both the generator and discriminator improve.

Deep Learning for Vision Systems by Mohamed Elgendy (2020)

GAN Minimax function

- minimax game theory
    - Zero-sum game
    - increase in one player’s score results in a decrease in another player’s score.

- The goal of the discriminator (*D*) is to maximize the probability of getting the correct label of the image.
- The generator’s *(G)* goal, on the other hand, is to minimize the chances of getting caught.

$$\underset{G}{\mathrm{Min}}\ \underset{D}{\mathrm{Max}}\ V(D,G) = E_{x\sim p_{data}}[\underbrace{\log D(x)}_{\substack{\textbf{Discriminator output}\\ \textbf{for real data } x}}] + E_{z\sim P_{z(z)}}[\log(1-\underbrace{D(G(z))}_{\substack{\textbf{Discriminator output}\\ \textbf{for generated fake data } G(z)}})]$$

Evaluating GAN Model

- Inception Score:
- Measures both the **quality** and **diversity** of images generated by a GAN.

- Fréchet inception distance (FID)
- Compares the **real** and **generated** image distributions in a shared feature space to assess how close the generated data is to real data.

References

- Guide to CNNs for CV – Khan et al. (2018)
- Deep Learning with Python – Chollet (2018)
- Deep Learning in Computer Vision – Awad & Hassaballah (2020)

- Deep Learning for Vision Systems by Mohamed Elgendy (2020)