Udacity Connect Session

Research Paper and links

March 7, 2018

Do you recognise these celebrities?

Image Source: geek.com


DEEPFAKES
LOIS LANE

Generative Adversarial Networks

"This (GANs), and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion."

Yann LeCun, Director of AI Research at Facebook, Prof at NYU

Founding father of CNNs

Probability Distribution Functions

Uniform Distribution

You can interpret the PDF as going over the input space horizontally with the vertical axis showing the probability that some value occurs.

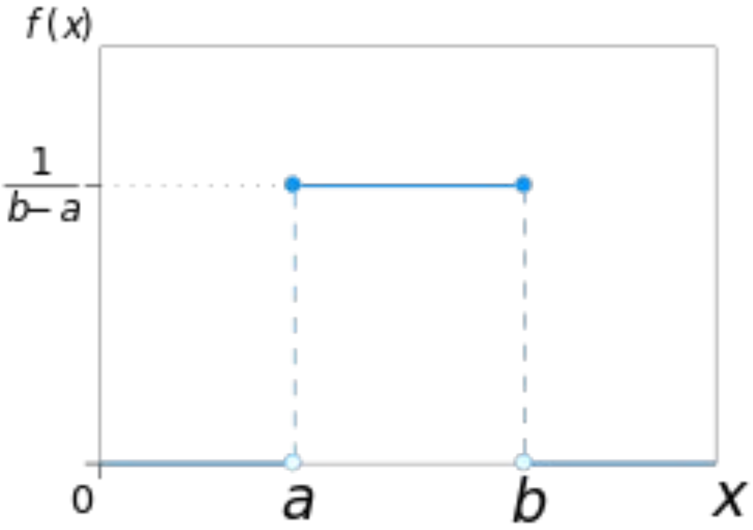


Image Source: [Wikipedia]

Probability Distribution Functions

Gaussian Distribution

Values near the mean are more likely than those far from it

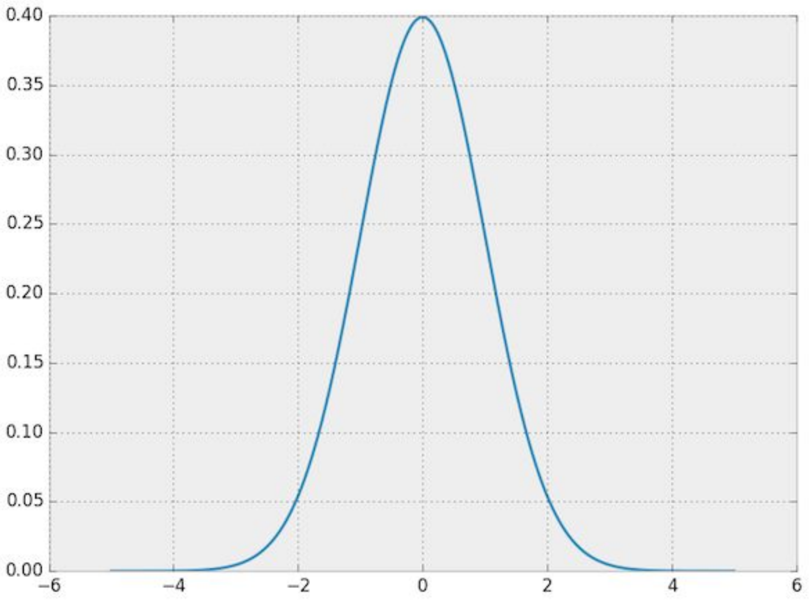


Image Source: [Image Completion with Deep Learning in TensorFlow]

Samples from Probability Distribution

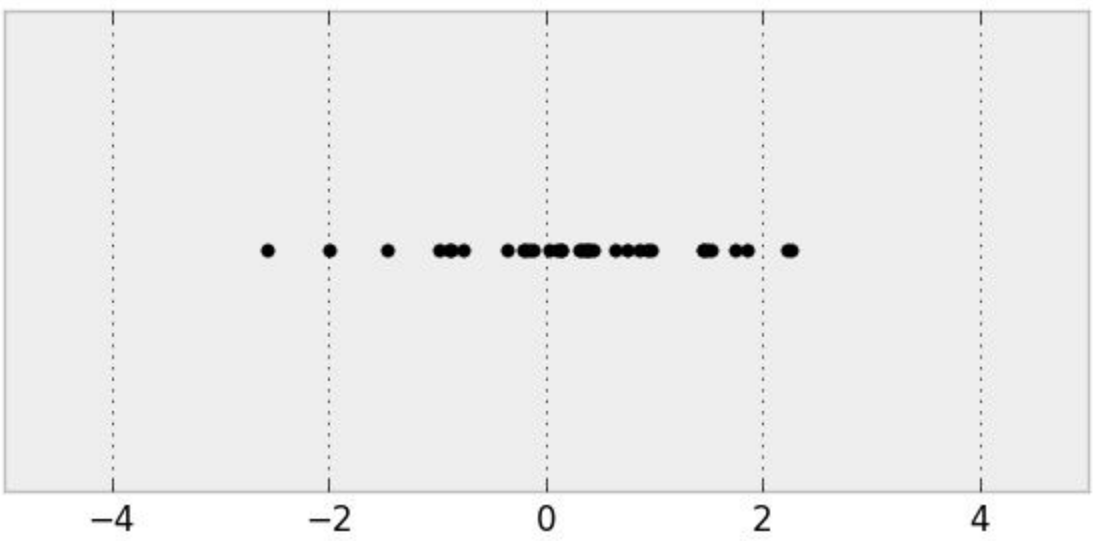


Image Source: [Image Completion with Deep Learning in TensorFlow]

Samples from Probability Distribution

2D Case

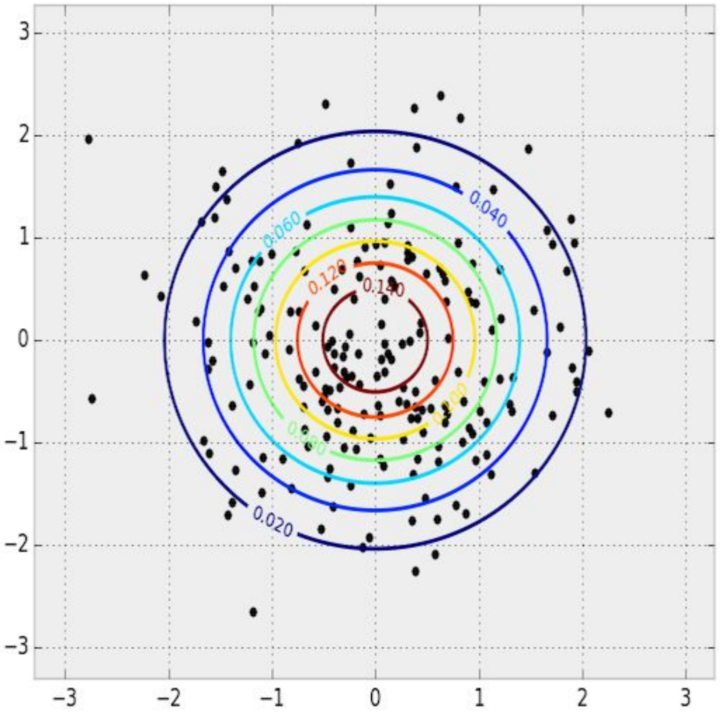
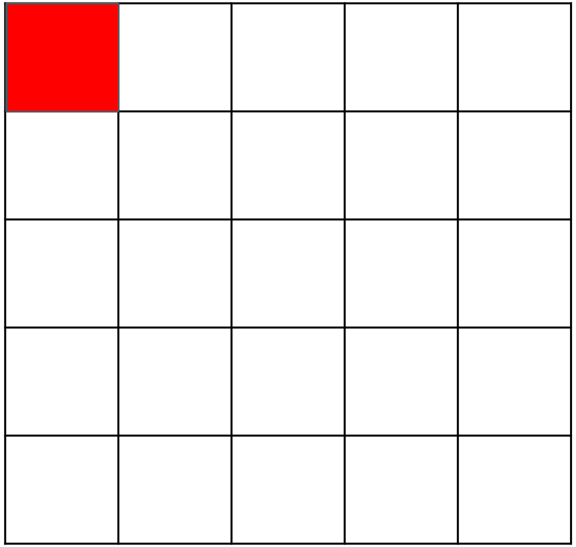


Image Source: [Image Completion with Deep Learning in TensorFlow]

Images as samples from probability distribution

Where does statistics fit in with images?



Do this for all 25 pixel locations. You will get 25 probability distributions

However pixels are not independent!

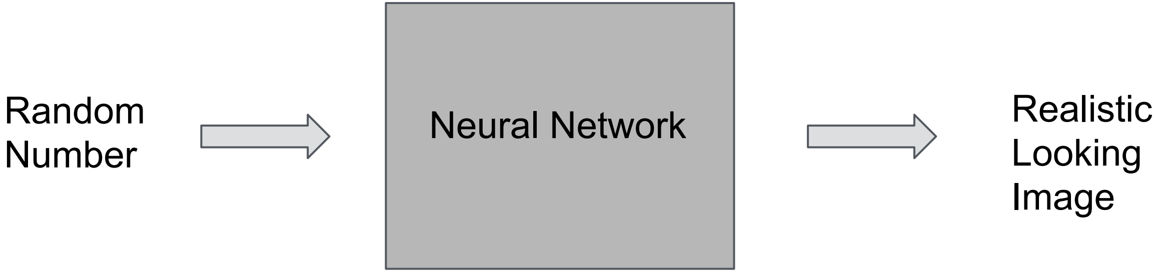
1. **Contextual information:** You can infer what missing pixels are based on information provided by surrounding pixels.
2. **Perceptual information:** You interpret the filled in portions as being “normal,” like from what you’ve seen in real life or from other pictures.

- The key relationship between images and statistics is that we can interpret images as samples from a high-dimensional probability distribution.
- If the images are 64x64 in size the the total dimensions ~= 64*64*3 = 12k

When you take an image with your camera, you are sampling from this complex probability distribution.

How to learn this high dimensional distribution?

Generative Networks



- Input can also be a vector of random numbers
- These numbers are sampled from a uniform distribution

Generative Networks

```
z = np.random.uniform(-1, 1, 5)
array([ 0.77356483,  0.95258473, -0.18345086,  0.69224724, -0.34718733])
```

```
def G(z):
   # TODO: implement this
   return imageSample


z = np.random.uniform(-1, 1, 5)
imageSample = G(z)
```

Down/Up Sampling

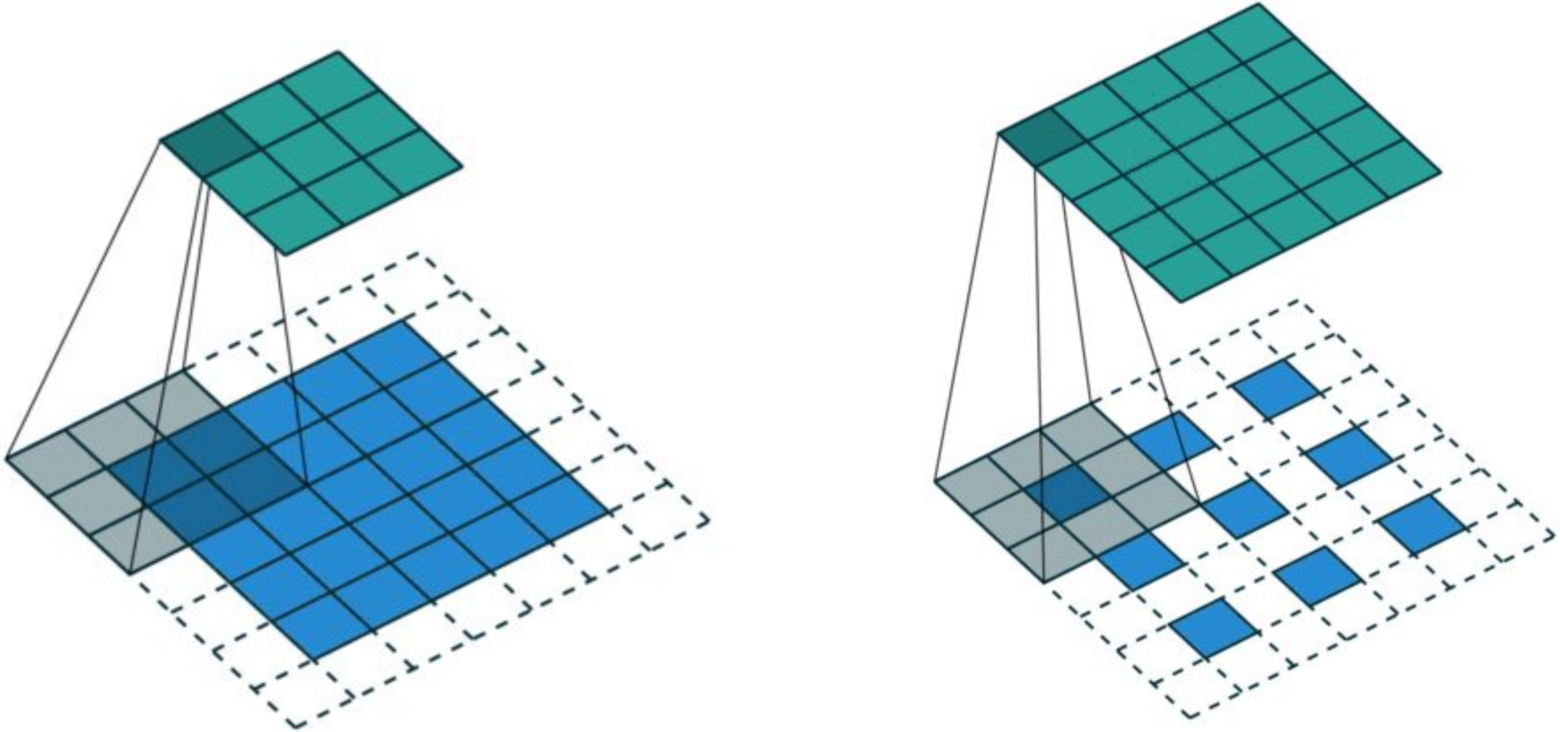


Image Source: github/vdumoulin/conv_arithmetic

DCGAN - Generator Architecture

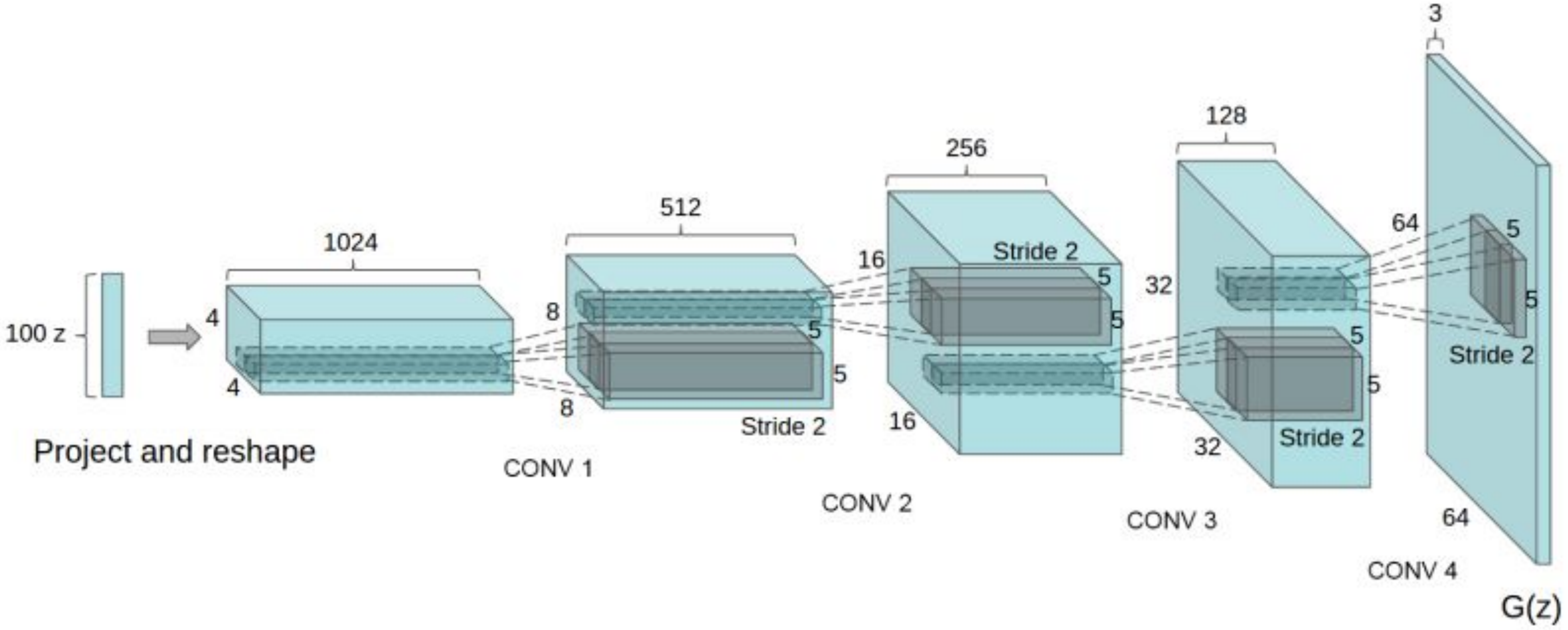


Image Source: [DCGAN Paper]()

Discriminators

DCGAN - Discriminator Architecture

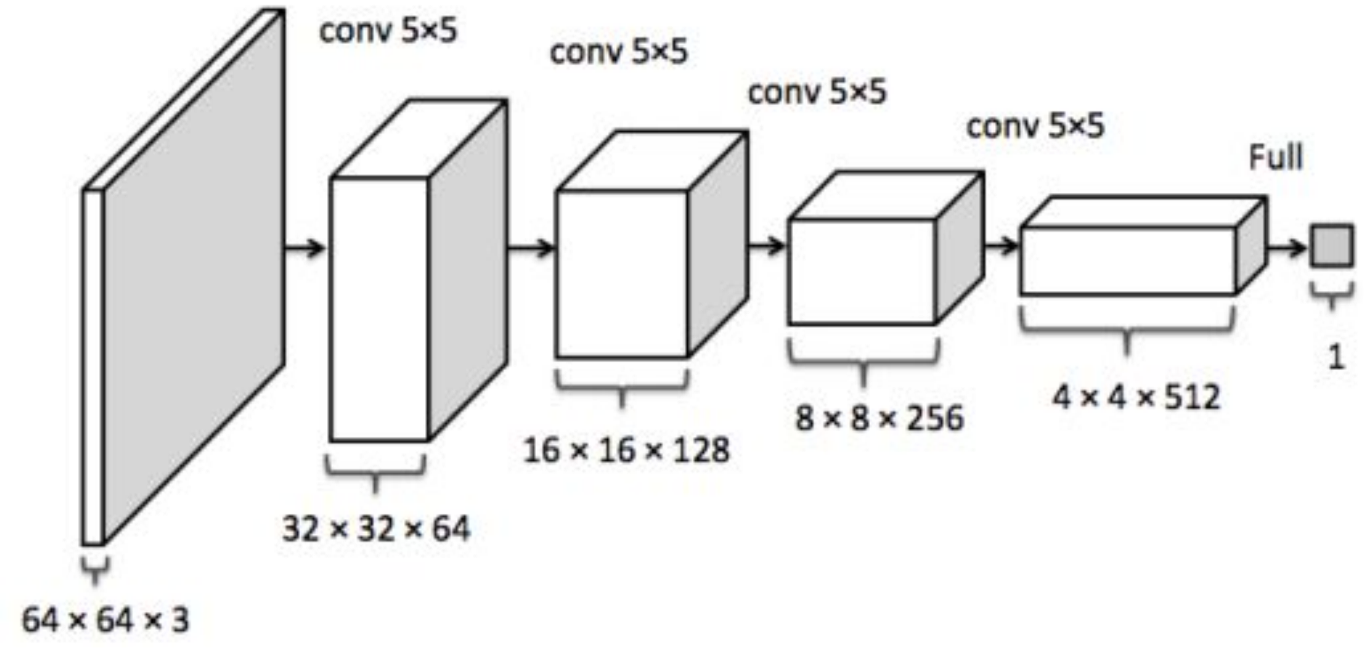


Image Source: [Image Completion with Deep Learning in TensorFlow]()

DCGAN - Combined Architecture

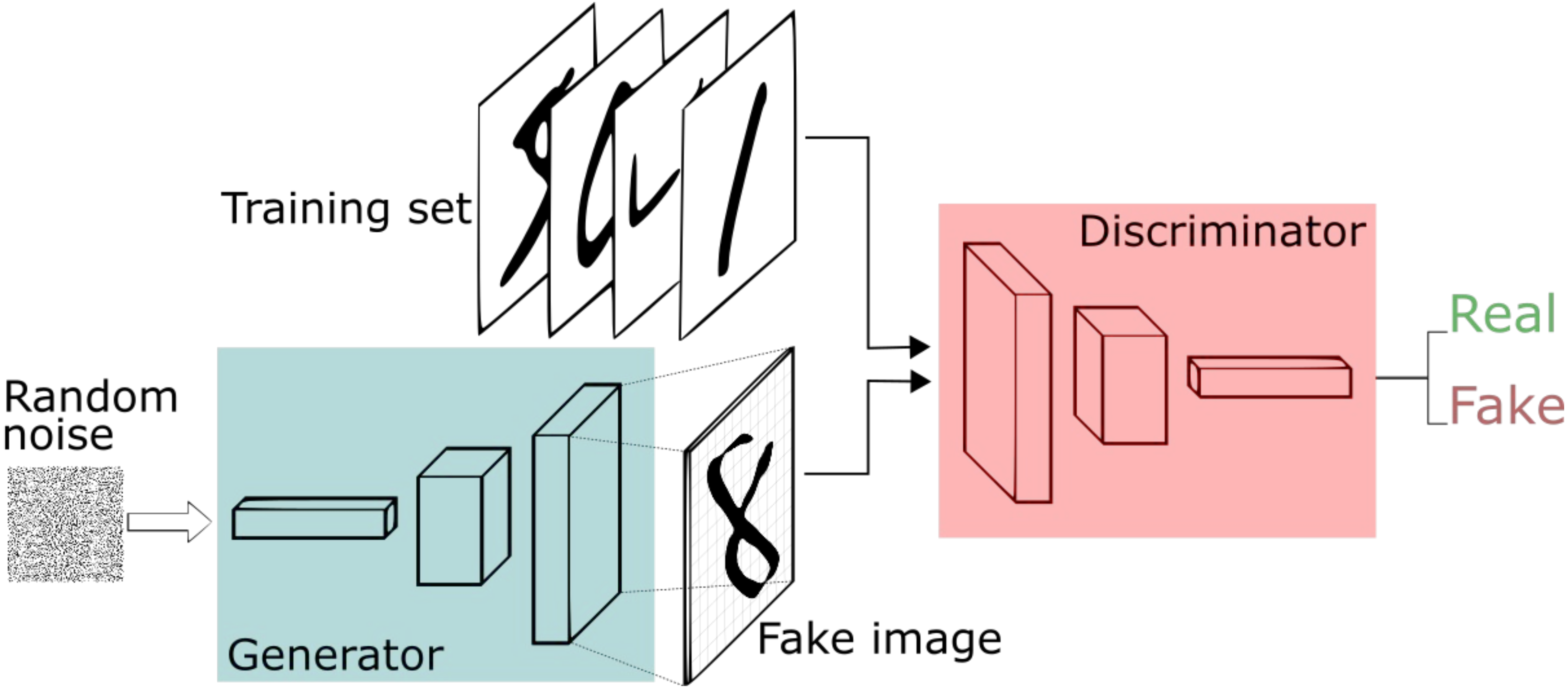


Image Source: deeplearning4j/generative-adversarial-network

DCGAN - Optimization Objective

- Assumptions
  - **x** is the real world input image
  - Discriminator: function **D**
  - **z**: input random vector
  - Generator: function **G**

- Discriminator
  - Maximise **D**(x)
  - Minimise **D**(**G**(z)), or maximise **1 - D**(**G**(**z**))
- Generator
  - Maximise **D**(**G**(z))
  - In other words minimise **1 - D**(**G**(**z**))

DCGAN - Optimization Objective

Probability Distribution Notation	Meaning
p_z	The (known, simple) distribution z goes over
p_{data}	The (unknown) distribution over our images. This is where our images are sampled from.
p_g	The generative distribution that the generator G samples from. We would like for $p_g = p_{\text{data}}$

Training DCGAN

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

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

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

lSource: [GAN Paper]

Results

Fake Images generated by the model

Fake images generated by model trained on LSUN bedrooms dataset. Image Source: [DCGAN Paper]

Results

Fake Images generated by the model

Fake images generated by model trained on human face dataset. Image Source: [DCGAN Paper]

Results

Fake Images generated by the model

Ground Truth MNIST GAN DCGAN

Image Source: [DCGAN Paper]

Variations of GANs

github:eriklindernoren/Keras-GAN

Auxiliary Classifier GAN	Bidirectional GAN
Adversarial Autoencoder	Boundary-Seeking GAN
Conditional GAN	Context-Conditional GAN
Context Encoder	Coupled GANs
CycleGAN	Deep Convolutional GAN
DualGAN	Generative Adversarial Network
InfoGAN	LSGAN
Pix2Pix	Semi-Supervised GAN
Super-Resolution GAN	Wasserstein GAN

Applications of GANs

Image to Image Translation

Using a variation called Cycle GAN

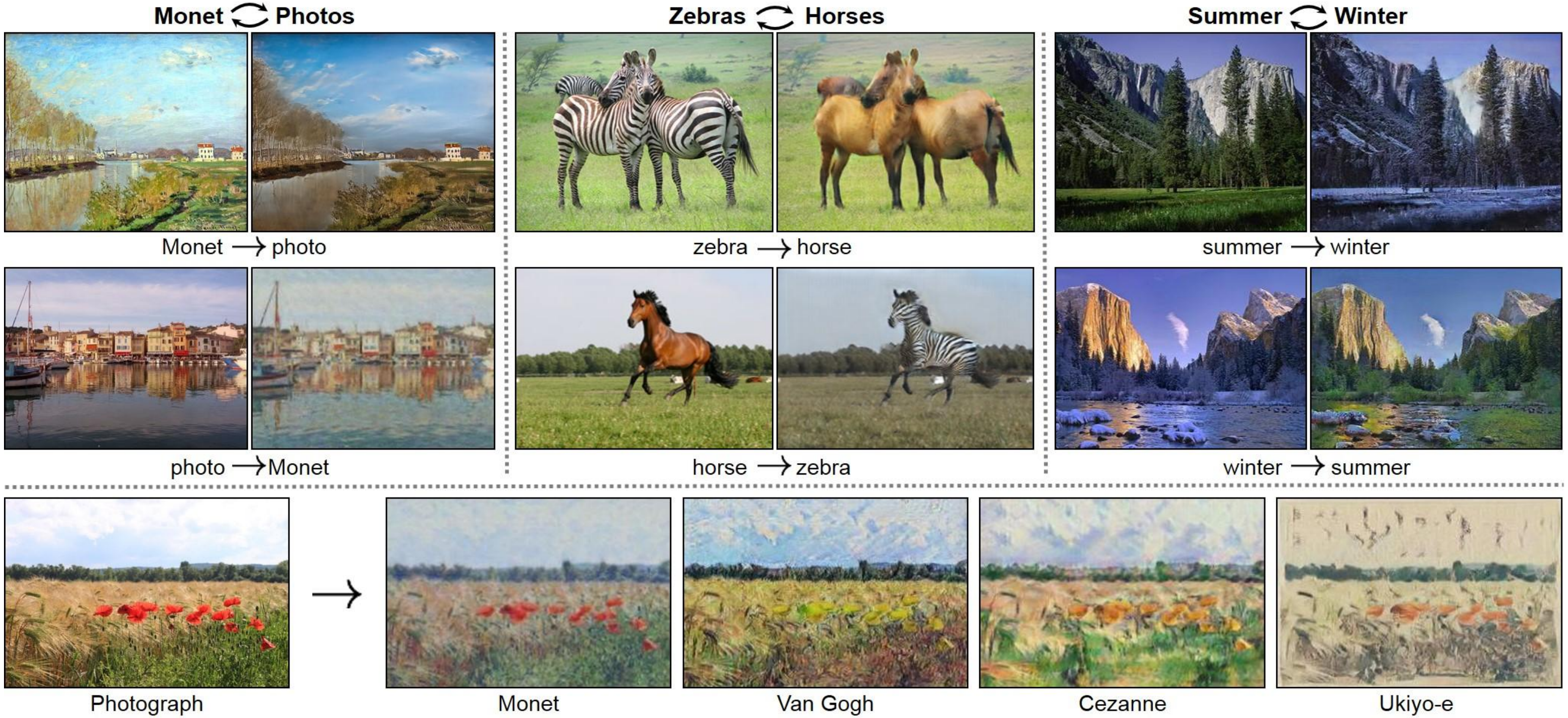


Image Source: github:junyanz/CycleGAN

Image to Image Translation

Using a variation called Cycle GAN

Image Source: [github:junyanz/CycleGAN](https://github.com/junyanz/CycleGAN)

Image Super-resolution

Using a variation called SR GAN

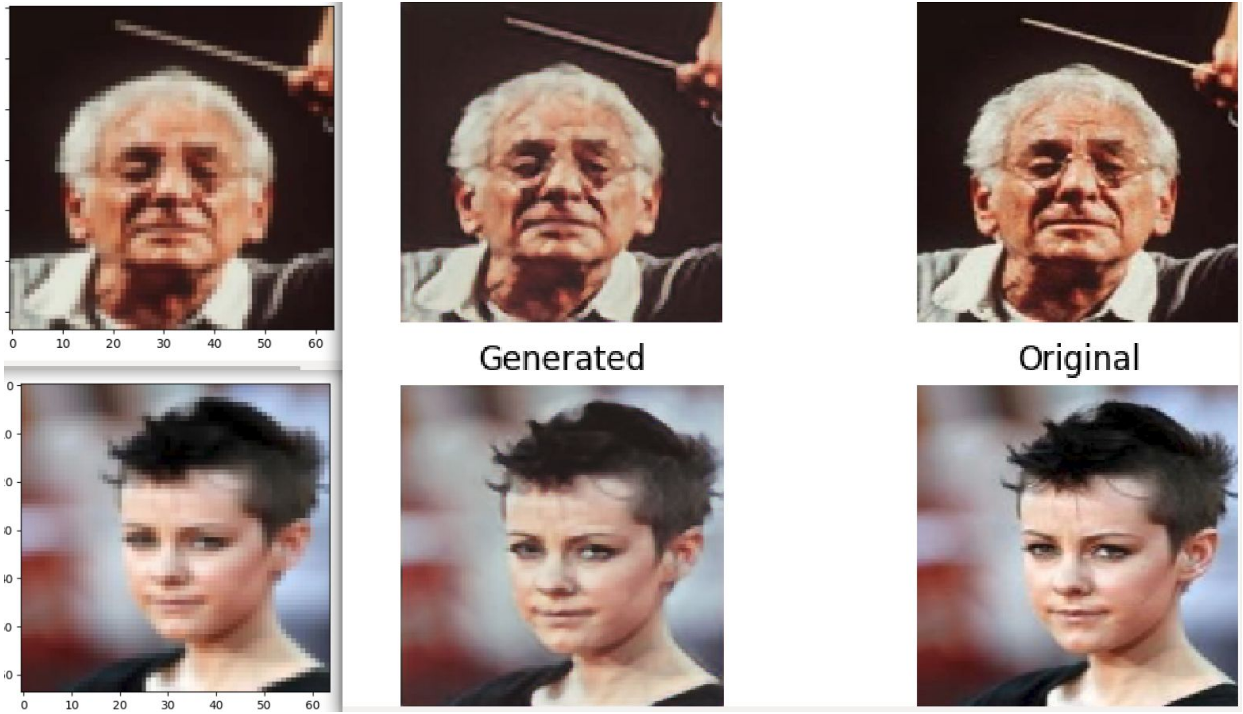


Image Source: github:eriklindernoren/Keras-GAN

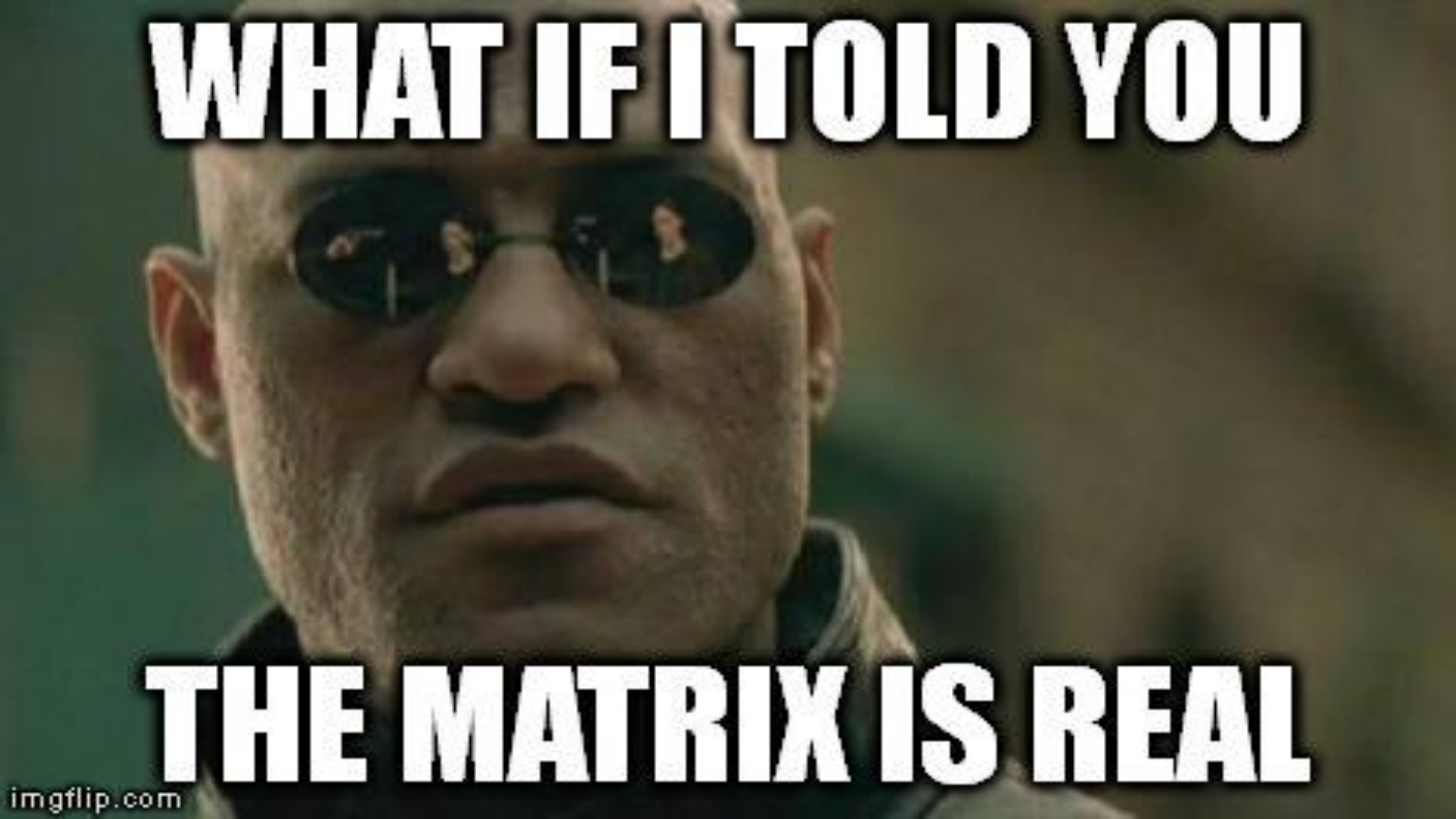
Semi Supervised Learning with GANs

1. Lets say you have a lot of unlabelled data and very small labelled dataset
2. Train a GAN with the unlabelled data
3. Then modify the discriminator to produce an additional output indicating the label of the input.
4. Train the discriminator on the labelled dataset
5. OpenAI's implementation gives close to state of the art results on the MNIST dataset, with only 10 labelled images per class. More info
6. The state of the art algorithm trains on 60,000 labelled images
7. So GANs have the potential to reduce the requirements for labelled dataset

Image Source: github:eriklindernoren/Keras-GAN

SimGAN: Generating Data

[SimGANs - a game changer in unsupervised learning, self driving cars, and more]


WHAT IF I TOLD YOU
THE MATRIX IS REAL
imgflip.com

Acknowledgement

1. Content is heavily borrowed from this blog post - Image Completion with Deep Learning in TensorFlow
2. Also this one - https://deeplearning4j.org/generative-adversarial-network

UDACITY

Be in Demand