

# Generative Adversarial Networks

## Deep Learning — Unit 9

Dr. Jon Krohn

`jon@untapt.com`

Slides available at `jonkrohn.com/talks`

December 1st, 2018

# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# Outline

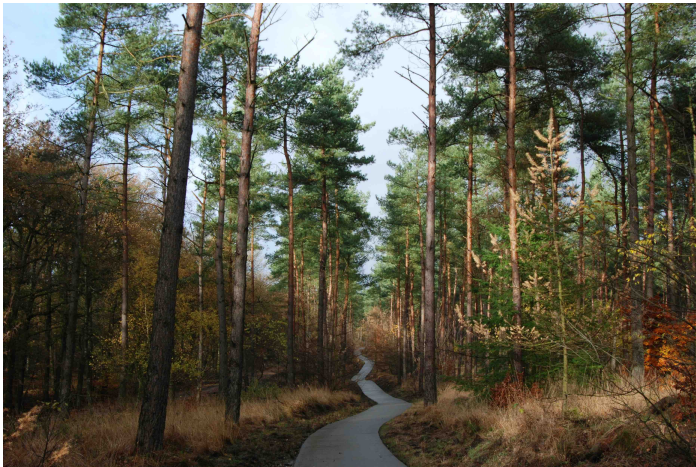
- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# Progress Check

## Your Deep Learning Project V



# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark



# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Progress Check

## Your Deep Learning Project V

Where are you at with respect to the following?

### 1 Splitting your data

- training set (80% — for optimizing parameters)
- validation set (10% — for hyperparameters)
- test set (10% — don't touch yet!)

### 2 Building and assessing architecture

- get above chance (simplifying problem, if necessary)
- do existing performance benchmarks exist?
- if not, use a simple architecture as benchmark

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post



# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

# Recommended Projects

## Your Deep Learning Project V

- 1 build a machine-vision architecture to classify images, e.g.:
  - [Fashion MNIST]
  - one of dozens of “image” data sets from [CrowdFlower]
  - one of the *Computer Vision* data sets from [Luke de Oliveira’s post]
- 2 build a natural language processing architecture to classify text, e.g.:
  - Yelp or Amazon sentiment [datasets] from [Zhang et al.]
  - the Yahoo! answers categories data set from Zhang et al.
  - one of dozens of “sentiment” or “text” data sets from CrowdFlower
  - one of the *Natural Language* data sets from Luke de Oliveira’s post

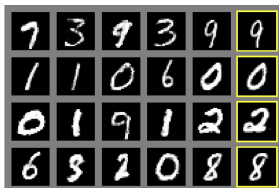


# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

# GANs

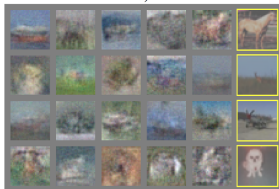
Goodfellow et al. (2014)



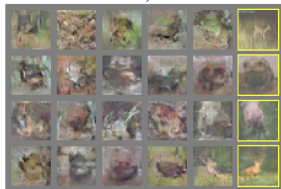
a)



b)



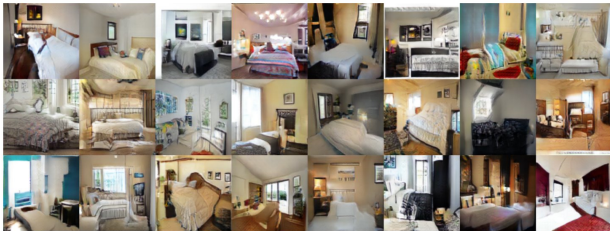
c)



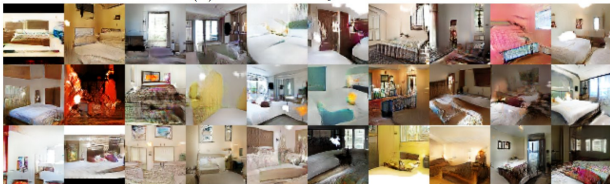
d)

# DCGANs

Radford et al. (2016)



(a) Generated by LSGANs.

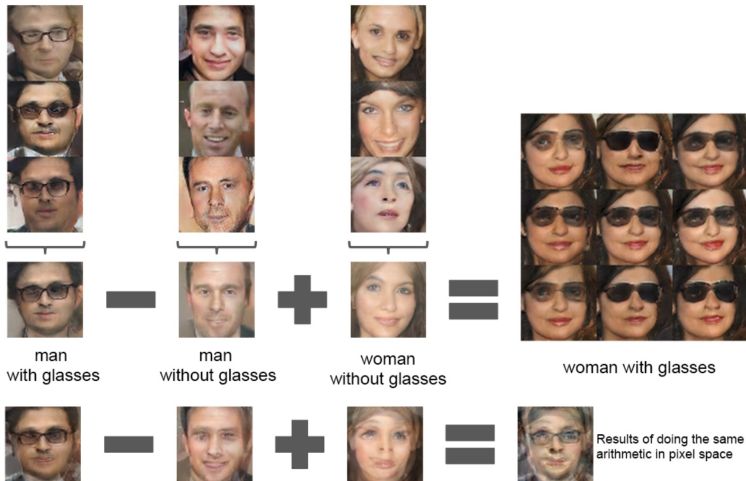


(b) Generated by DCGANs (Reported in [13]).

Figure 5: Generated images on LSUN-bedroom.

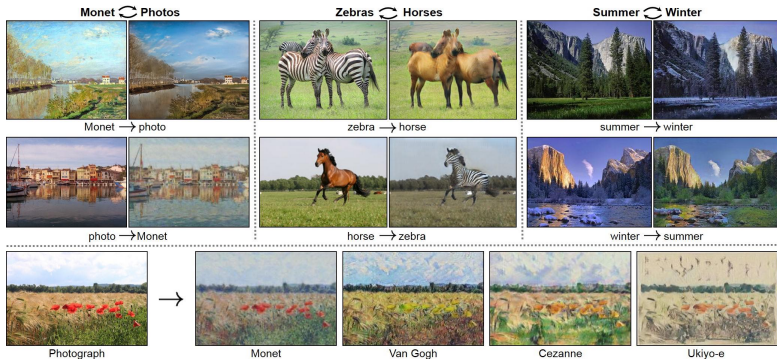
# DCGANs

Radford et al. (2016)



# CycleGANs

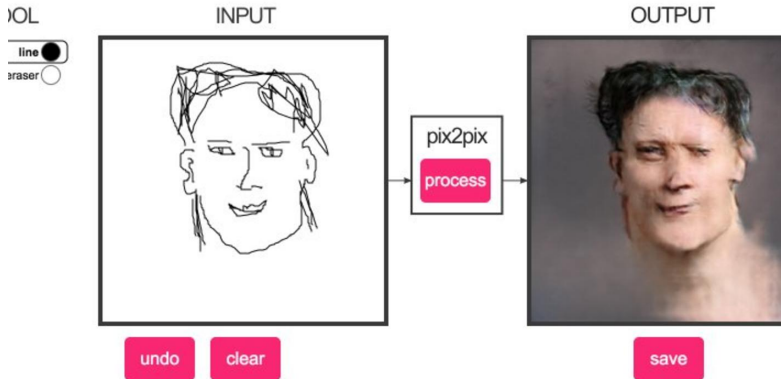
Zhu et al. (2017)



<https://junyanz.github.io/CycleGAN>

# pix2pix

Isola et al. (2017)



<https://>

# StackGAN

Zhang et al. (2017)



Figure 3. Example results by our proposed StackGAN, GAWWN [20], and GAN-INT-CLS [22] conditioned on text descriptions from CUB test set. GAWWN and GAN-INT-CLS generate 16 images for each text description, respectively. We select the best one for each of them to compare with our StackGAN.

[“celebrity” latent-space interpolation]



# Latent-Space Interpolation

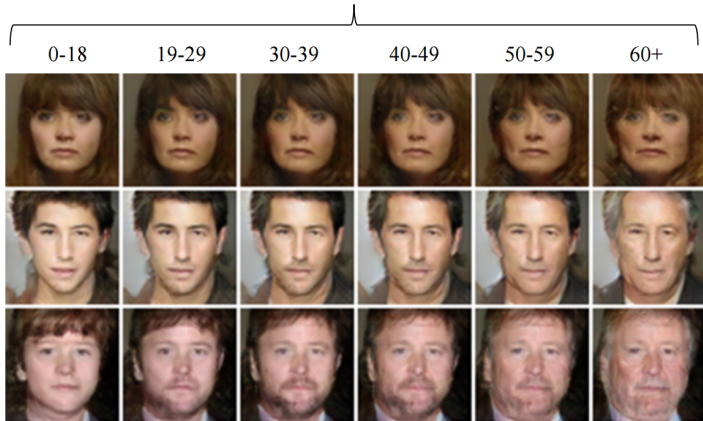
Your Projects

Applications

Theory

In Practice

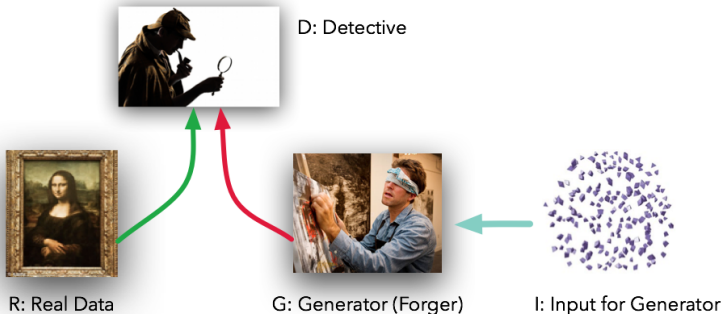
Face Aging



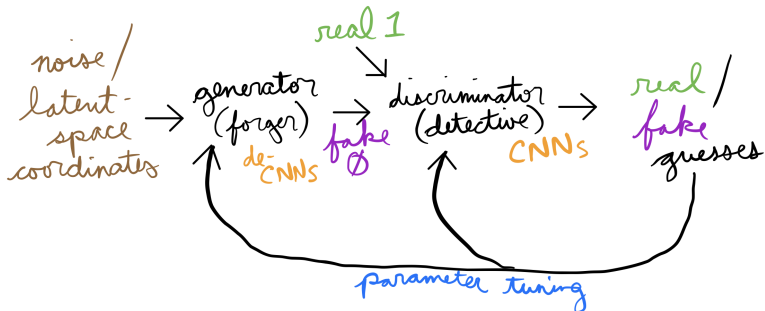
# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory**
- 4 “Quick, Draw!” Implementation

# Goodfellow et al. (2014)



# Goodfellow et al. (2014)



# 1-D Gaussian

## Approximating a Toy Distribution

[ video ]

# Outline

- 1 Deep Learning Projects
- 2 Applications
- 3 Essential Theory
- 4 “Quick, Draw!” Implementation

[Quick, Draw!]

# GANimation

(Requires Adobe Acrobat Reader)



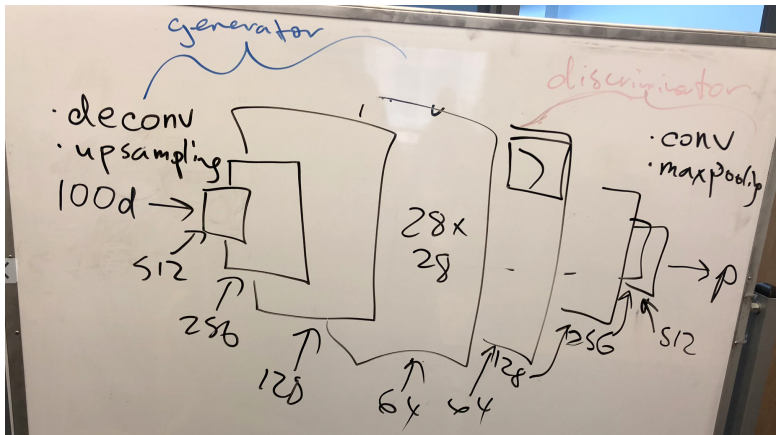
# GAN Code

Your Projects

Applications

Theory

In Practice



[ notebook ]