Go to **www.menti.com** and use the code **53 39 35**

CS230: Lecture 5
Attacking Networks with Adversarial Examples
-
Generative Adversarial Networks

Kian Katanforoosh

Today's outline

I. Attacking NNs with Adversarial Examples

II. Generative Adversarial Networks

I. Adversarial examples

Discovery (2014): several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples

A. How to build adversarial examples and attack a network?

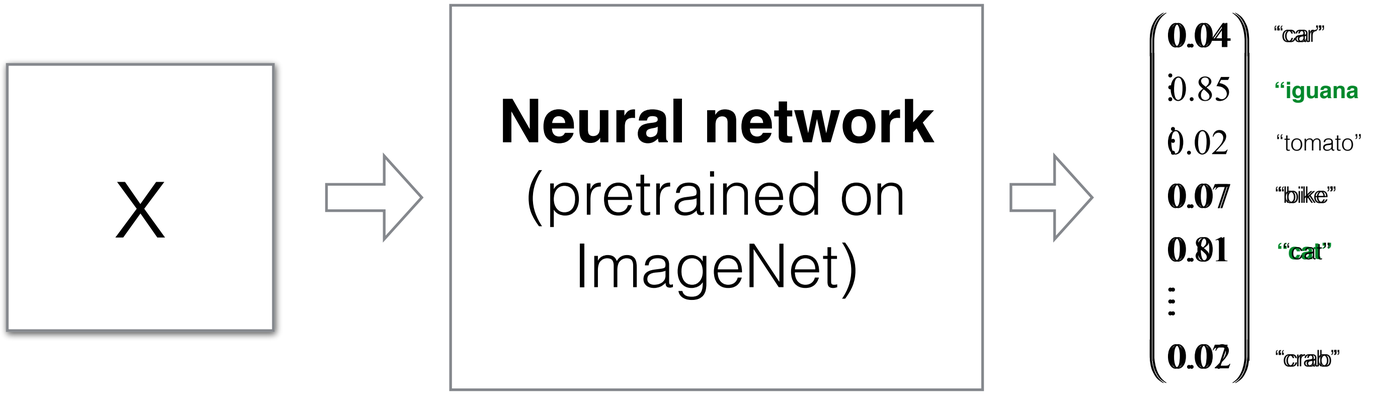
B. Examples

C. How to defend against adversarial examples?

[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

I. A. How to build adversarial examples and attack a network?

Goal: Given a pretrained network on ImageNet, find an example that is not a iguana but will be classify as an iguana.



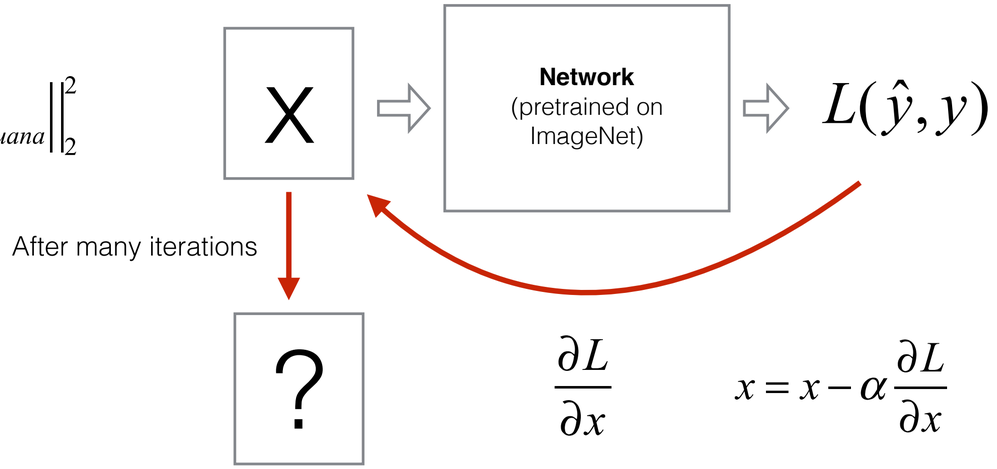
1. Rephrasing what we want:

Find x such that: $\hat{y}(x) = y_{iguana} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

2. Defining the loss function

$$L(\hat{y}, y) = \frac{1}{2}\left\| \hat{y}(W, b, x) - y_{iguana} \right\|_2^2$$

3. Optimize the image

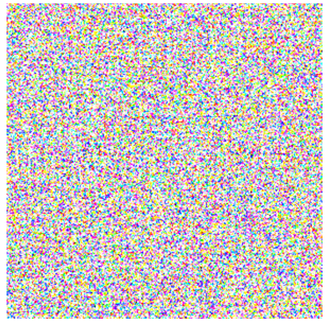


[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

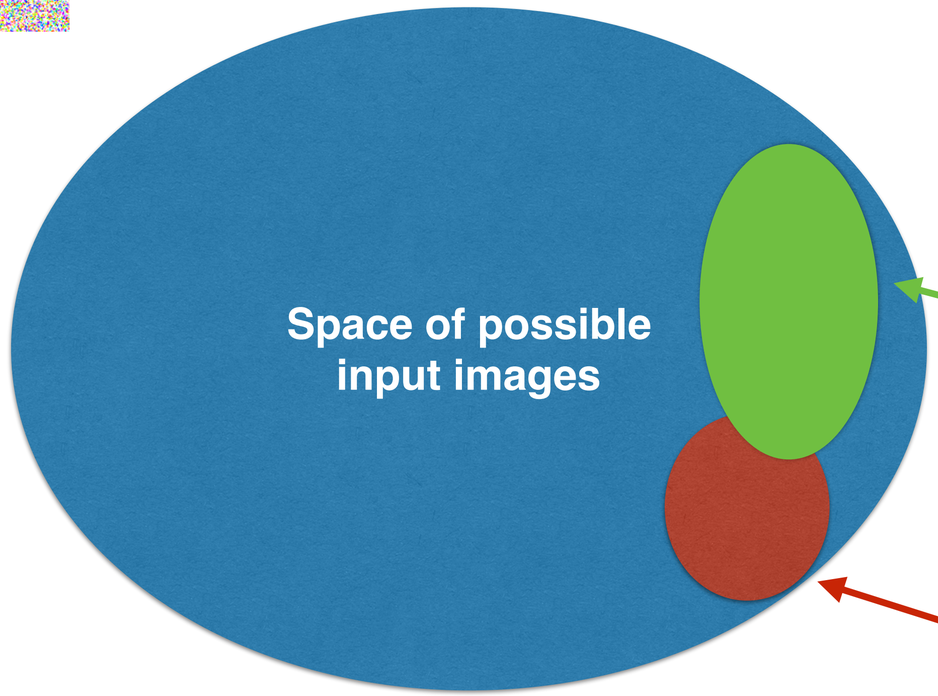
I. A. How to build adversarial examples and attack a network?

Question: Will the learned image **x** look like an **iguana**?



?

$$256^{32\times32\times3} \approx 10^{7400}$$



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. A. How to build adversarial examples and attack a network?

Goal: Given a pretrained network on ImageNet, find an example that is a cat but will be classify as an iguana.

Neural network (pretrained on ImageNet)

$$\begin{pmatrix} 0.04 \\ 0.85 \\ 0.02 \\ 0.07 \\ 0.81 \\ \vdots \\ 0.02 \end{pmatrix} \begin{matrix} \text{“car”} \\ \text{“iguana”} \\ \text{“tomato”} \\ \text{“bike”} \\ \text{“cat”} \\ \\ \text{“crab”} \end{matrix}$$

1. Rephrasing what we want:

Find x such that: $\hat{y}(x) = y_{iguana} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

And: $x = x_{cat}$

2. Defining the loss function

$$L(\hat{y}, y) = \frac{1}{2}\left\|\hat{y}(W, b, x) - y_{iguana}\right\|_2^2$$

$$+\lambda\left\|x - x_{cat}\right\|_2^2$$

3. Optimize the image

Network (pretrained on ImageNet)

$L(\hat{y}, y)$

After many iterations

$$\frac{\partial L}{\partial x} \qquad x = x - \alpha \frac{\partial L}{\partial x}$$

[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

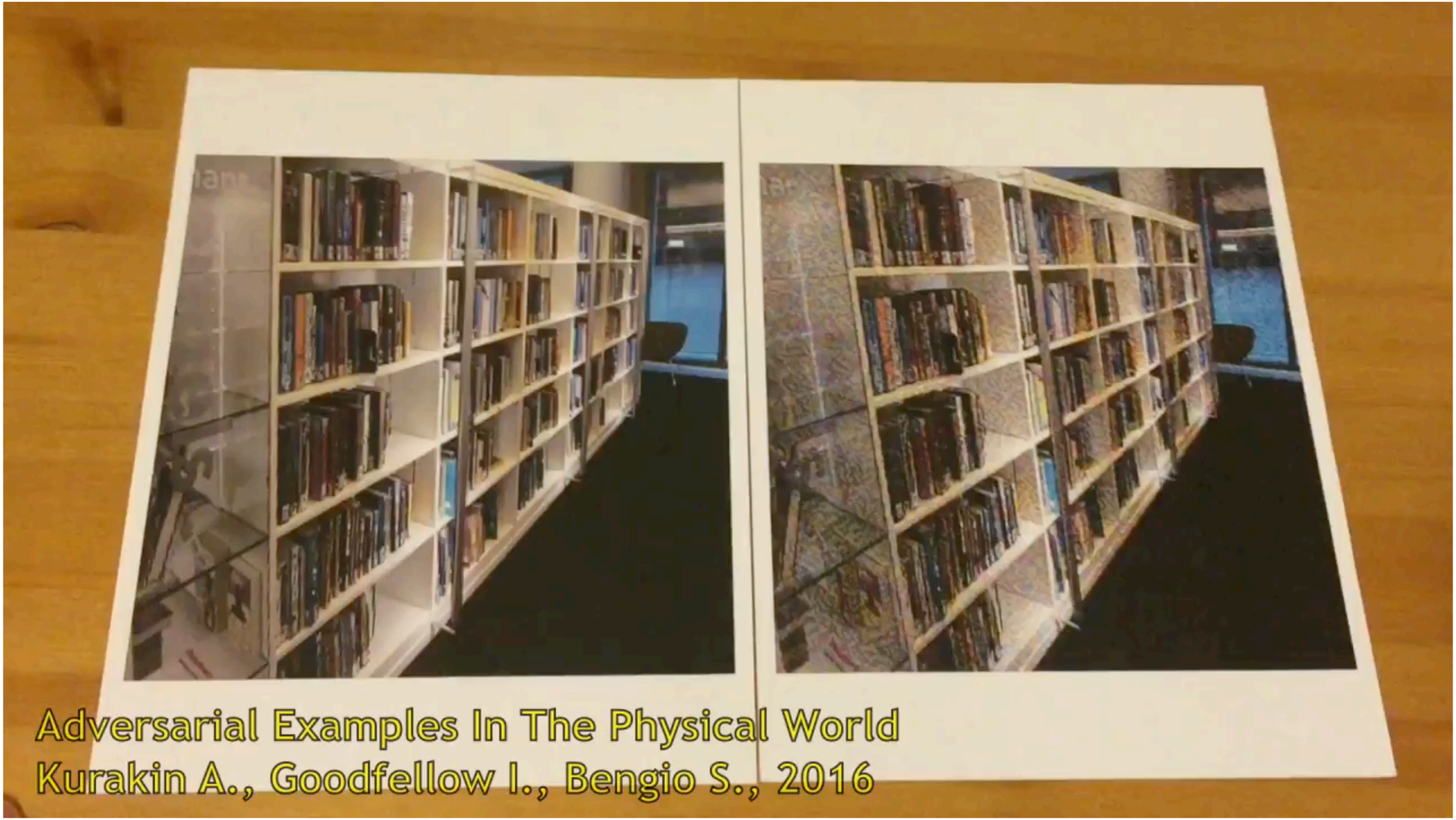
Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. A. How to build adversarial examples and attack a network?

92% Cat

94% Iguana

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri



[Alexey Kurakin, Ian J. Goodfellow, Samy Bengio (2017): Adversarial examples in the physical world]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. B. How to defend against adversarial examples?

Types of attacks:

- Non-targeted attacks
- Targeted attacks

Knowledge of the attacker:

- White-box
- Black-box

Solution 1

- Train on correctly labelled adversarial examples

x = 

y = cat

Solution 2

- Adversarial training $L_{new} = L(W, b, x, y) + \lambda L(W, b, x_{adv}, y)$
- Adversarial logit pairing $L_{new} = L(W, b, x, y) + \lambda \left|\left| f(x; W, b) - f(x_{adv}; W, b) \right|\right|_2^2$

Do neural networks actually understand the data?

[Harini Kannan et al. (2018): Adversarial Logit Pairing]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II. Generative Adversarial Networks (GANs)

A. Motivation

B. G/D Game

C. Practical tips to train/evaluate GANs

D. Interesting results

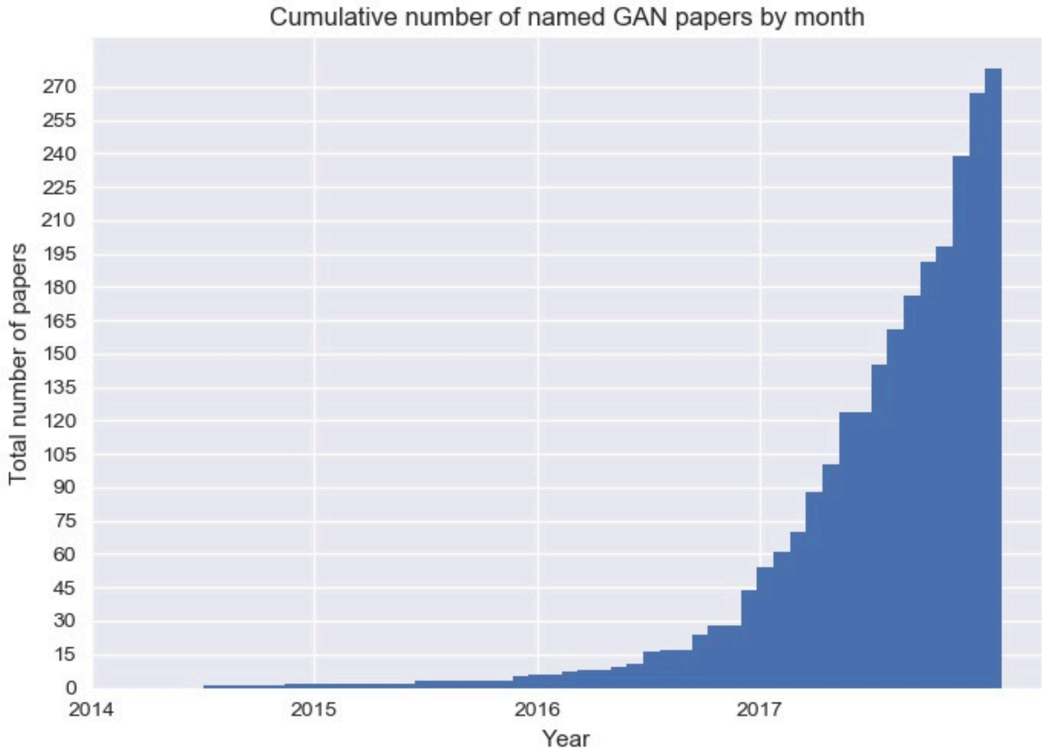
[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

II.A - Motivation

Motivation: endowing computers with an understanding of our world.

Goal: collect a lot of data, use it to train a model to generate similar data from scratch.

Intuition: number of parameters of the model << amount of data



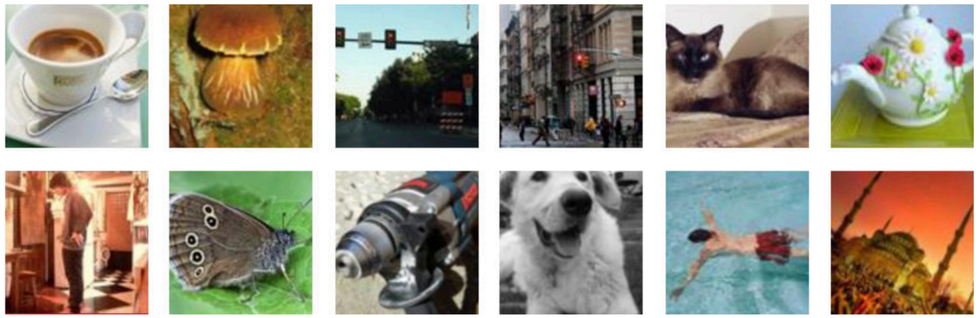
[The Gan Zoo (2017)]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

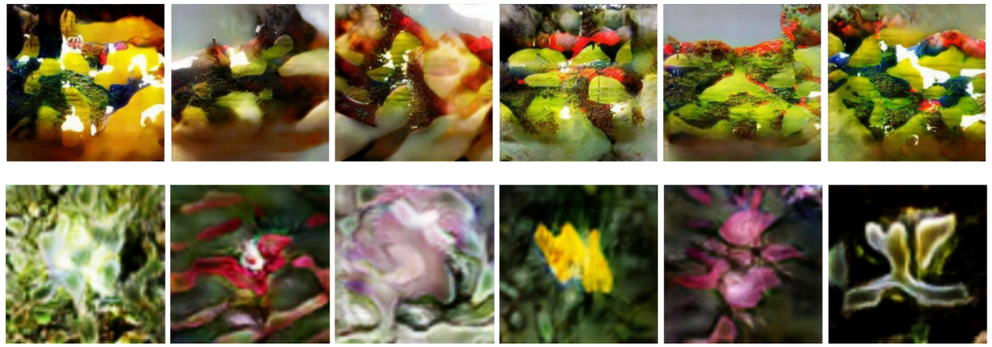
II.A - Motivation

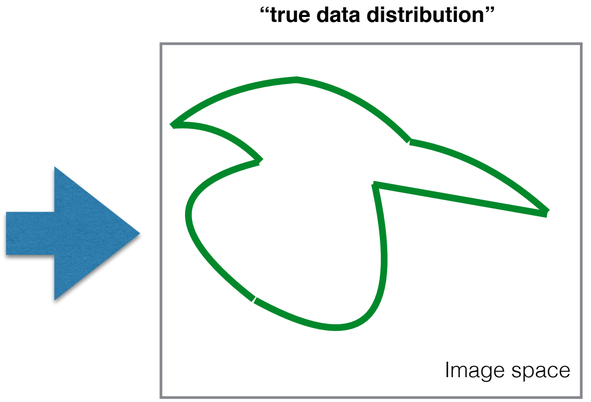
Probability distributions:

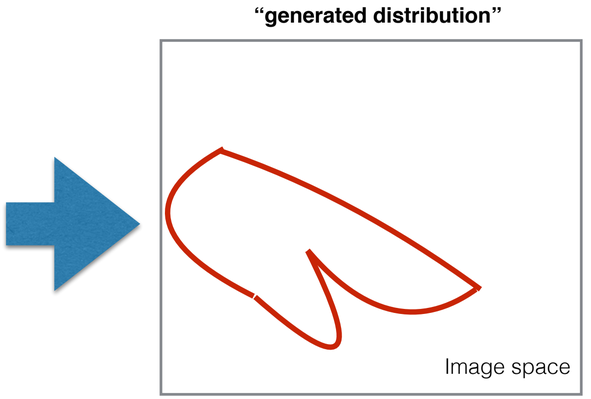
Samples from the “true data distribution”



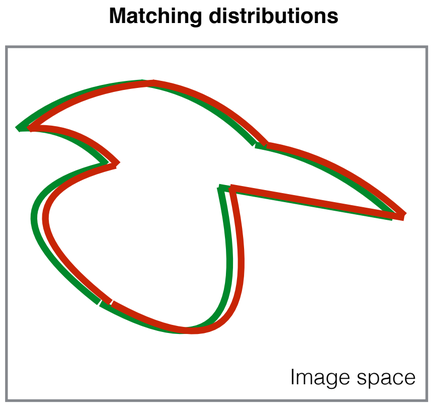
Samples from the “generated distribution”







Goal



[Andrej Karpathy et al. (2016): Generative Models, OpenAI blog]

II.B - G/D Game

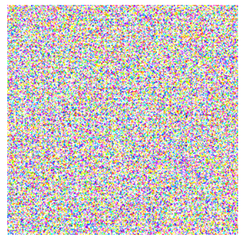
100-d
random code

$$\begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$$

z

**Generator “G”
(Neural Network)**

(64,64,3)
generated image



$\neq$

How can we train G to generate images from the true data distributions?

[Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas (2017): StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks]

II.B - G/D Game

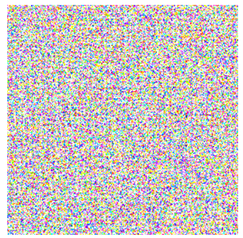
100-d
random code

$$\begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$$

z

**Generator "G"
(Neural Network)**

(64,64,3)
generated image



Run Adam simultaneously on two minibatches
(true data / generated data)

x

**Discriminator "D"
(Neural Network)**

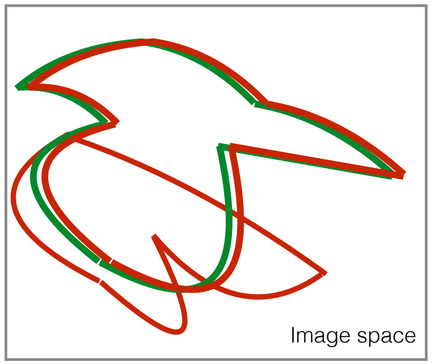
Gradients

Binary classification

$$\begin{cases} y = 0 \quad if \quad x = G(z) \\ y = 1 \quad otherwise \end{cases}$$

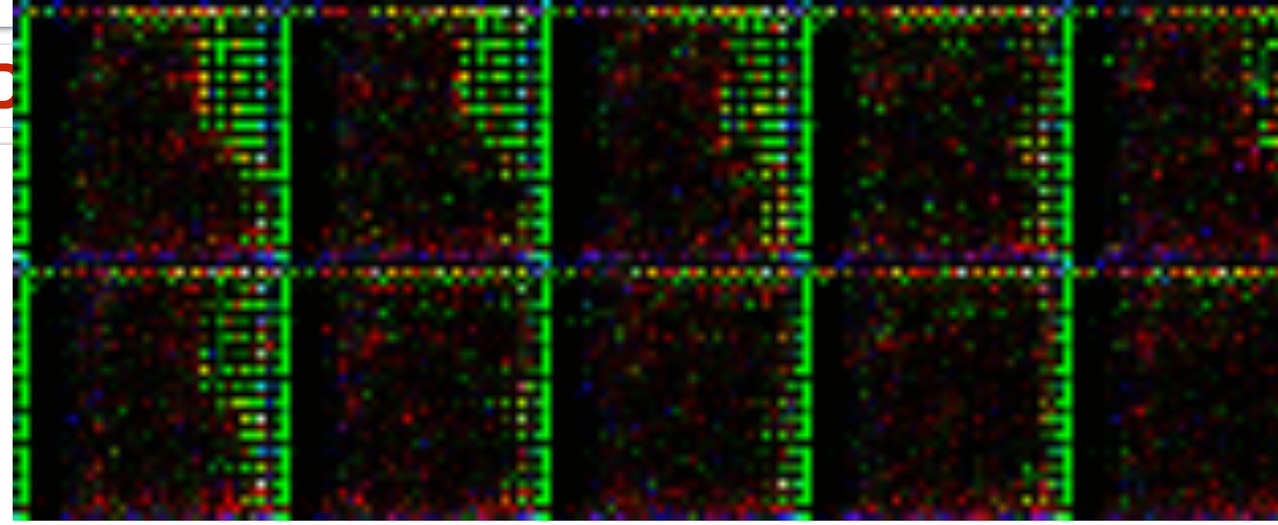
**Real images
(database)**

Probability distributions



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.B - G/D



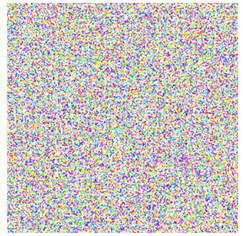
100-d
random code

$$\begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$$

z

**Generator "G"
(Neural Network)**

(64,64,3)
generated image



End goal: G is outputting images that are indistinguishable from real images for D

x

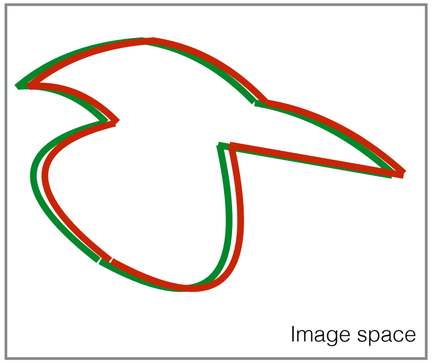
**Discriminator "D"
(Neural Network)**

Gradients Binary classification

$$\begin{cases} y = 0 \quad if \quad x = G(z) \\ y = 1 \quad otherwise \end{cases}$$

**Real images
(database)**

Probability distribution



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.B - G/D Game

Training procedure, we want to minimize:

Labels: $\begin{cases} y_{real} \text{ is always 1} \\ y_{gen} \text{ is always 0} \end{cases}$

- The loss of the discriminator

$$J^{(D)} = \underbrace{-\frac{1}{m_{real}} \sum_{i=1}^{m_{real}} y_{real}^{(i)} . \log(D(x^{(i)}))}_{\text{cross-entropy 1}} \underbrace{- \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} (1 - y_{gen}^{(i)}) . \log(1 - D(G(z^{(i)})))}_{\text{cross-entropy 2}}$$

cross-entropy 1:
"D should correctly label real data as 1"

cross-entropy 2:
"D should correctly label generated data as 0"

- The loss of the generator

$$J^{(G)} = -J^{(D)} = \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} \log(1 - D(G(z^{(i)})))$$

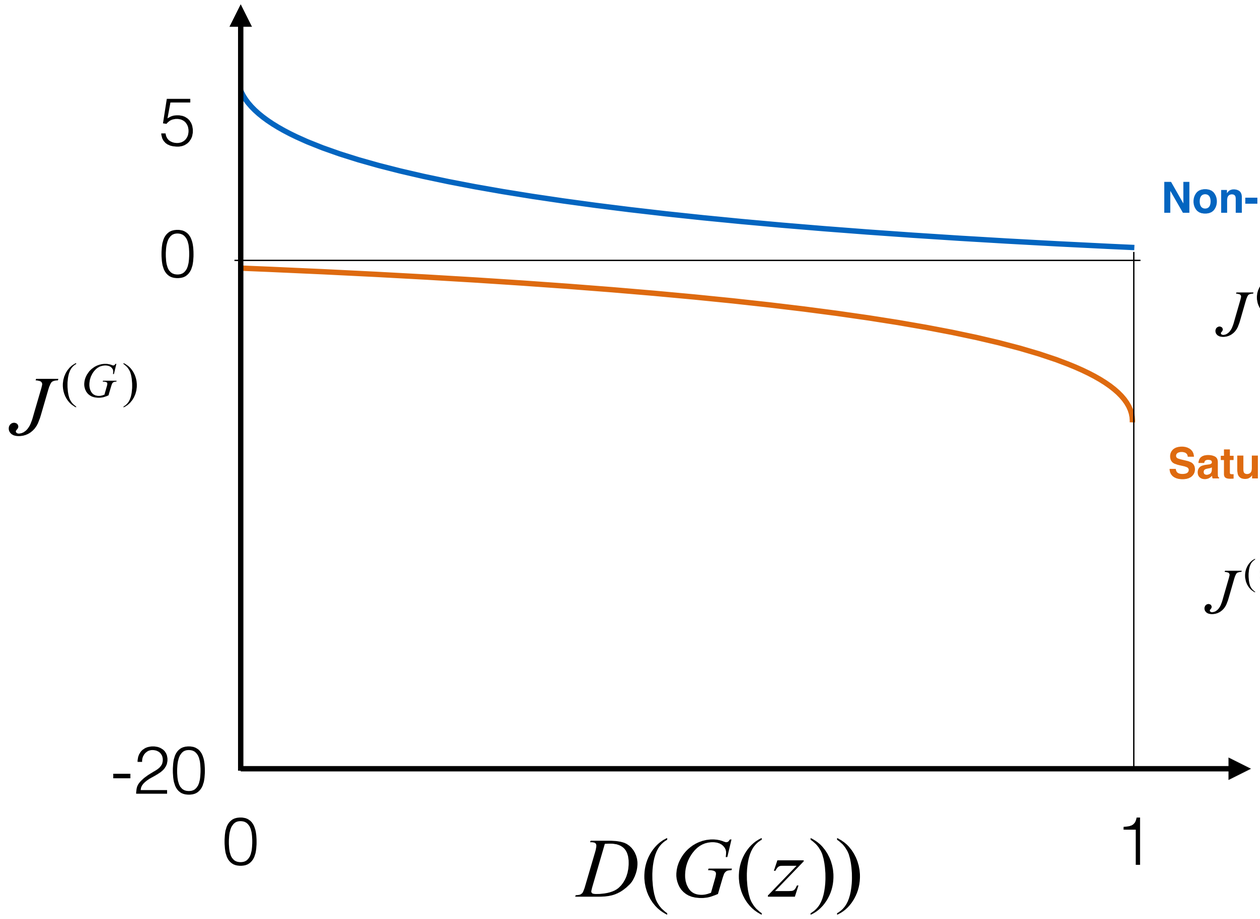
"G should try to fool D: by minimizing the opposite of what D is trying to minimize"

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.C - Training GANs

Saturating cost for the generator:

$$\min\left[\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(1-D(G(z^{(i)})))\right] \Leftrightarrow \max\left[\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))\right] \Leftrightarrow \min\left[-\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))\right]$$



Non-saturating cost

$$J^{(G)} = -\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))$$

Saturating cost

$$J^{(G)} = \frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(1-D(G(z^{(i)})))$$

[Ian Goodfellow (2014): NIPS Tutorial: GANs]

II.C - Training GANs

Note that: $$\min\left[\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(1-D(G(z^{(i)})))\right] \Leftrightarrow \max\left[\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))\right] \Leftrightarrow \min\left[-\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))\right]$$

New training procedure, we want to minimize:

$$J^{(D)} = \underbrace{-\frac{1}{m_{real}}\sum_{i=1}^{m_{real}} y_{real}^{(i)}.\log(D(x^{(i)}))}_{\text{cross-entropy 1:}} \underbrace{-\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}(1-y_{gen}^{(i)}).\log(1-D(G(z^{(i)})))}_{\text{cross-entropy 2:}}$$

cross-entropy 1:
"D should correctly label real data as 1"

cross-entropy 2:
"D should correctly label generated data as 0"

$$J^{(G)} = -\frac{1}{m_g}\sum_{i=1}^{m_g}\log(D(G(z^{(i)})))$$

"G should try to fool D: by minimizing this"

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

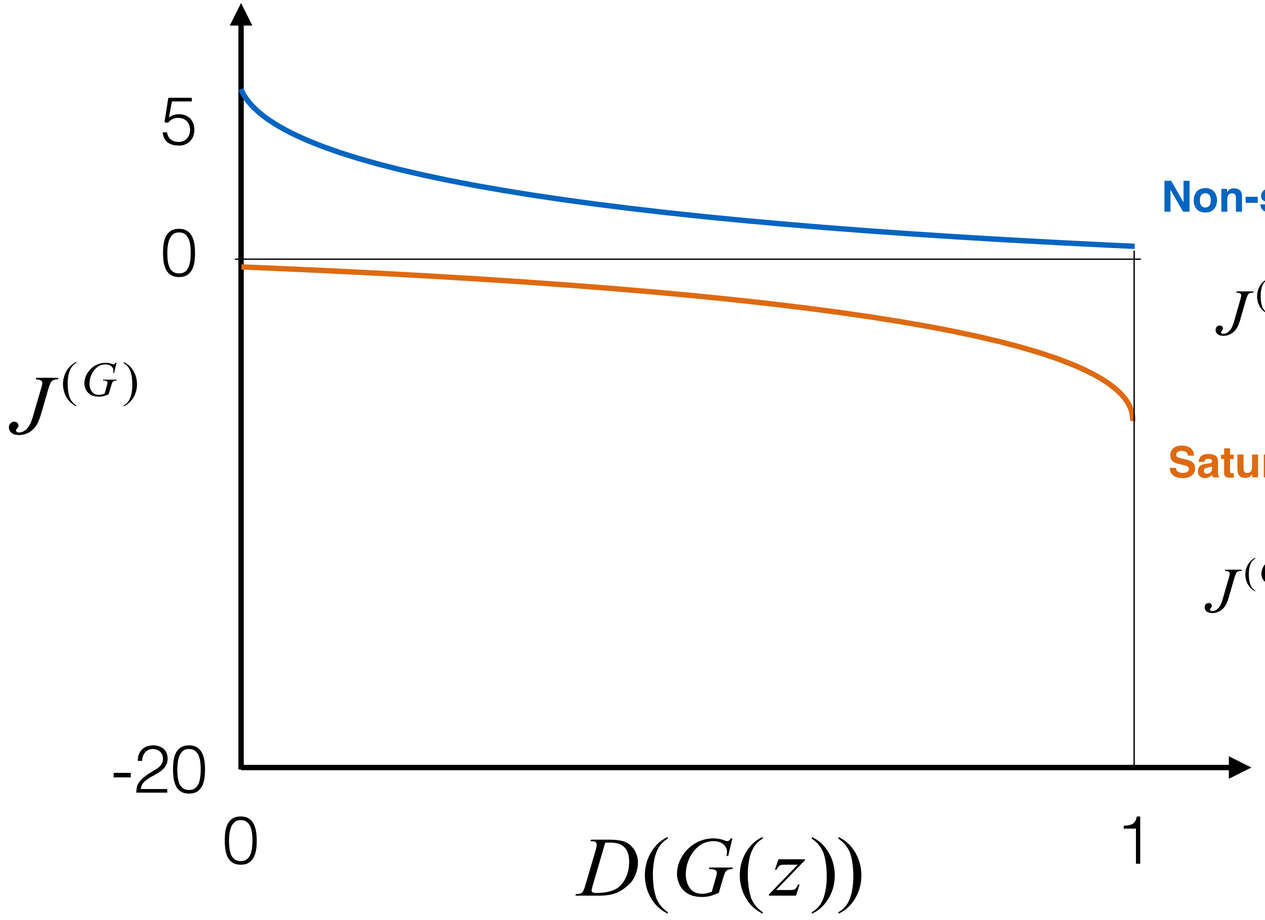
II.C - Training GANs

Simultaneously training G/D?

for num_iterations:

for k iterations:
update D

update G



Non-saturating cost

$$J^{(G)} = -\frac{1}{m_g}\sum_{i=1}^{m_g}\log(D(G(z^{(i)})))$$

Saturating cost

$$J^{(G)} = \frac{1}{m_g}\sum_{i=1}^{m_g}\log(1-D(G(z^{(i)})))$$

[Ian Goodfellow (2014): NIPS Tutorial: GANs]

II.C - Training GANs

BatchNorm with GANs:

Generated images (batch 1)

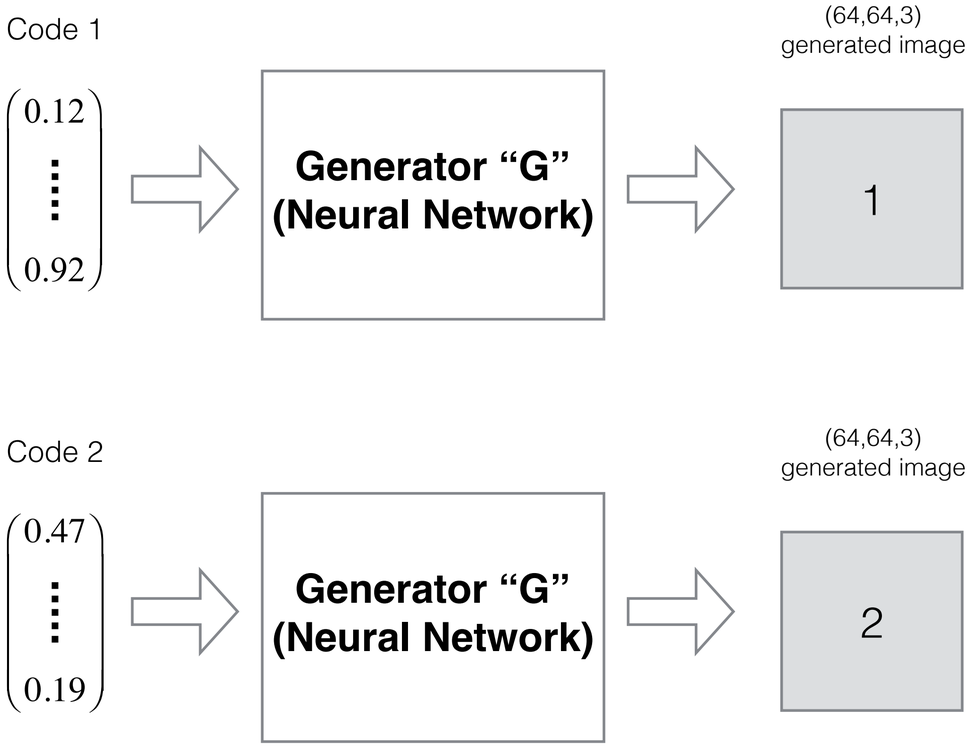
Generated images (batch 2)

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

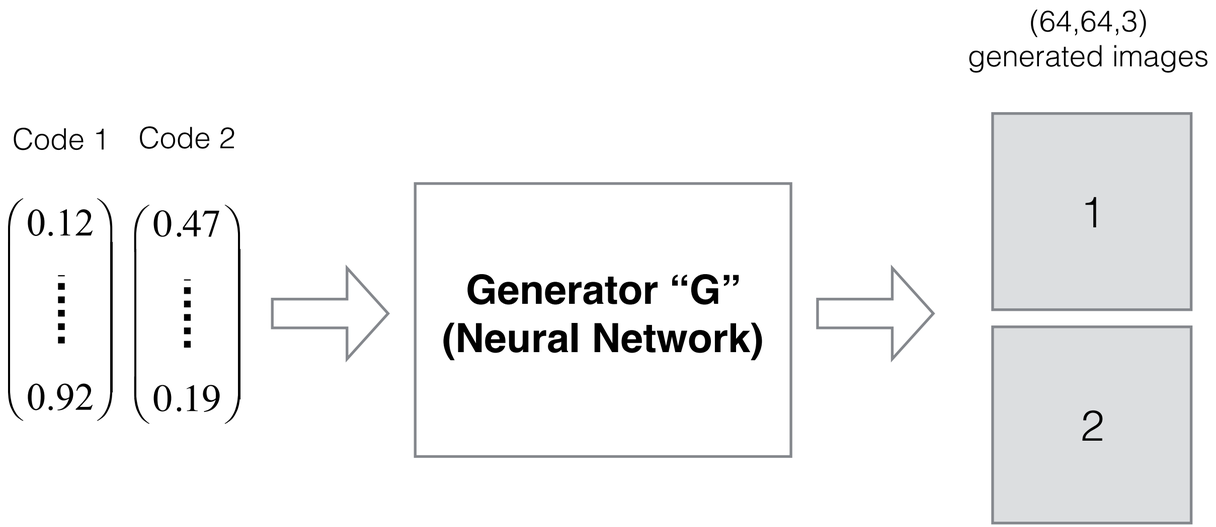
II.C - Training GANs

BatchNorm with GANs:

Assume no batchnorm



Assume batchnorm



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.C - Training GANs

BatchNorm with GANs:

BatchNorm

$$Z = \{z^{(1)}, ..., z^{(m)}\}$$

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} z^{(i)}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (z^{(i)} - \mu_B)^2$$

$$z_{norm}^{(i)} = \frac{z^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

$$\tilde{z}^{(i)} = \gamma z_{norm}^{(i)} + \beta$$

Reference BatchNorm

$$R = \{r^{(1)}, ..., r^{(m)}\}$$

$$Z = \{z^{(1)}, ..., z^{(m)}\}$$

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} r^{(i)}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (r^{(i)} - \mu_B)^2$$

$$z_{norm}^{(i)} = \frac{z^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

$$\tilde{z}^{(i)} = \gamma z_{norm}^{(i)} + \beta$$

Virtual BatchNorm

$$R = \{r^{(1)}, ..., r^{(m)}\}$$

$$Z = \{z^{(1)}, ..., z^{(m)}\}$$

For k = 1….m

$$\mu_B = \frac{1}{m+1}\left(z^{(k)} + \sum_{i=1}^{m} r^{(i)} \right)$$

$$\sigma_B^2 = \frac{1}{m+1}\left((z^{(k)} - \mu_B)^2 + \sum_{i=1}^{m} (r^{(i)} - \mu_B)^2 \right)$$

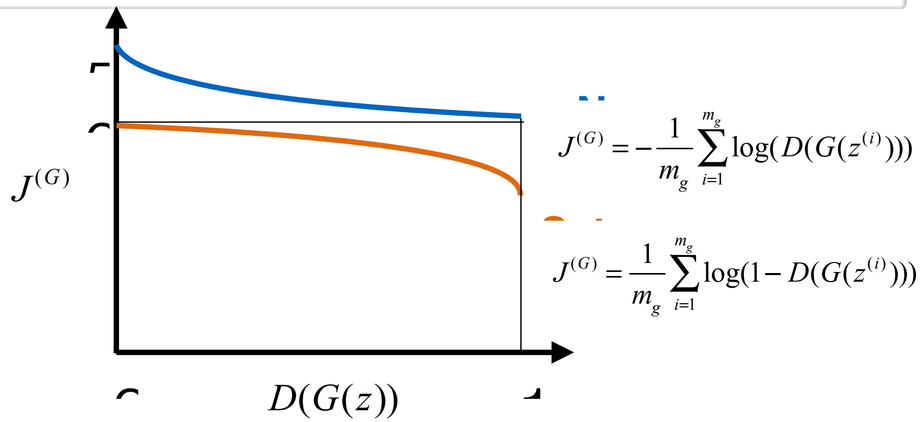
$$z_{norm}^{(k)} = \frac{z^{(k)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

$$\tilde{z}^{(k)} = \gamma z_{norm}^{(k)} + \beta$$

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.C - Training GANs

Recap: GANs' training tips



- Use the non-saturated cost function
- Keep D up-to-date with respect to G (k update for D / 1 update for G)
- Use Virtual Batchnorm
- (not presented but important) One-sided label smoothing

And a lot more, GANs are hard to train!

[Soumith et al. (2016): GanHacks]

II.D - Interesting results

Operation on codes

Code 1

$\begin{pmatrix} 0.12 \\ \vdots \\ 0.92 \end{pmatrix}$

Generator "G" (Neural Network)

(64,64,3) generated image

Code 2

$\begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$

Generator "G" (Neural Network)

(64,64,3) generated image

Code 3

$\begin{pmatrix} 0.42 \\ \vdots \\ 0.07 \end{pmatrix}$

Generator "G" (Neural Network)

(64,64,3) generated image

Code 1 Code 2 Code 3

$\begin{pmatrix} 0.12 \\ \vdots \\ 0.92 \end{pmatrix} - \begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix} + \begin{pmatrix} 0.42 \\ \vdots \\ 0.07 \end{pmatrix}$

Generator "G" (Neural Network)

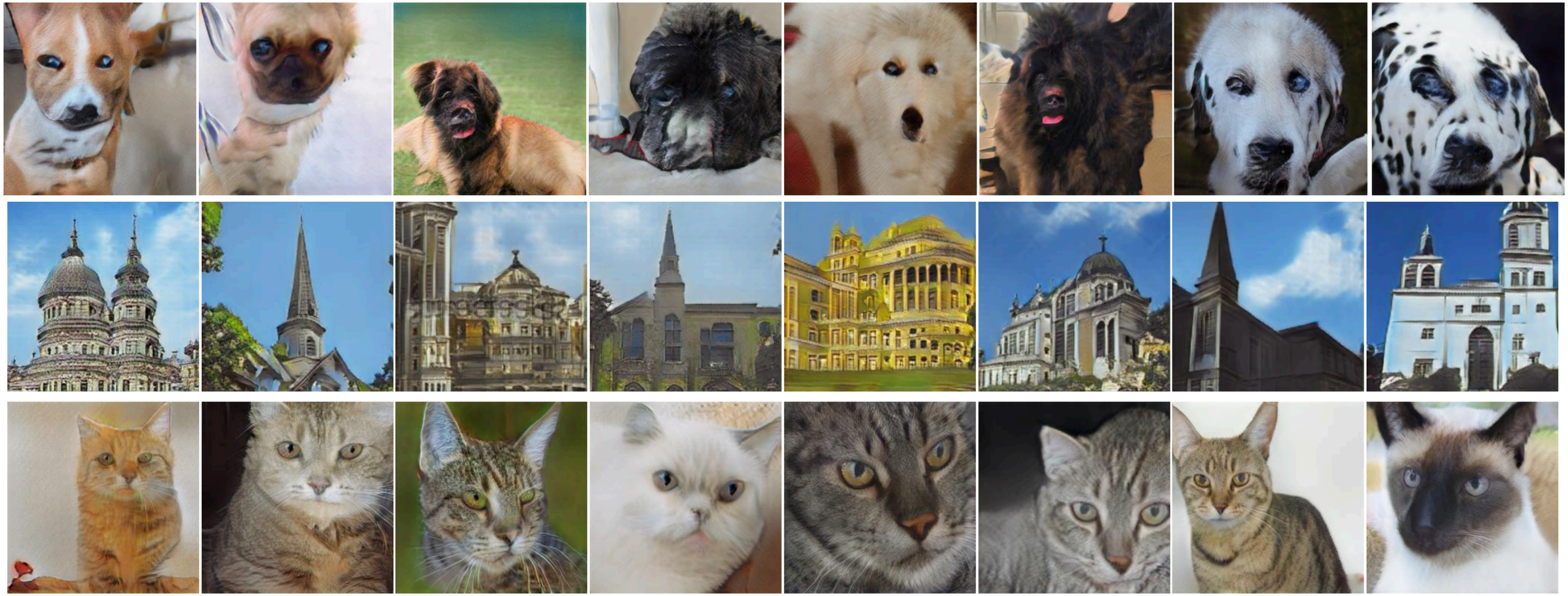
Man with glasses - man + woman = woman with glasses

[Radford et al. (2015): UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS]

II.D - Interesting results

Image Generation:

Samples from the “generated distribution”



[Zhang et al. (2017): StackGAN++]

II.D - Interesting results

Pix2Pix:

https://affinelayer.com/pixsrv/

[Isola et al. (2017): Image-to-Image Translation with Conditional Adversarial Networks]

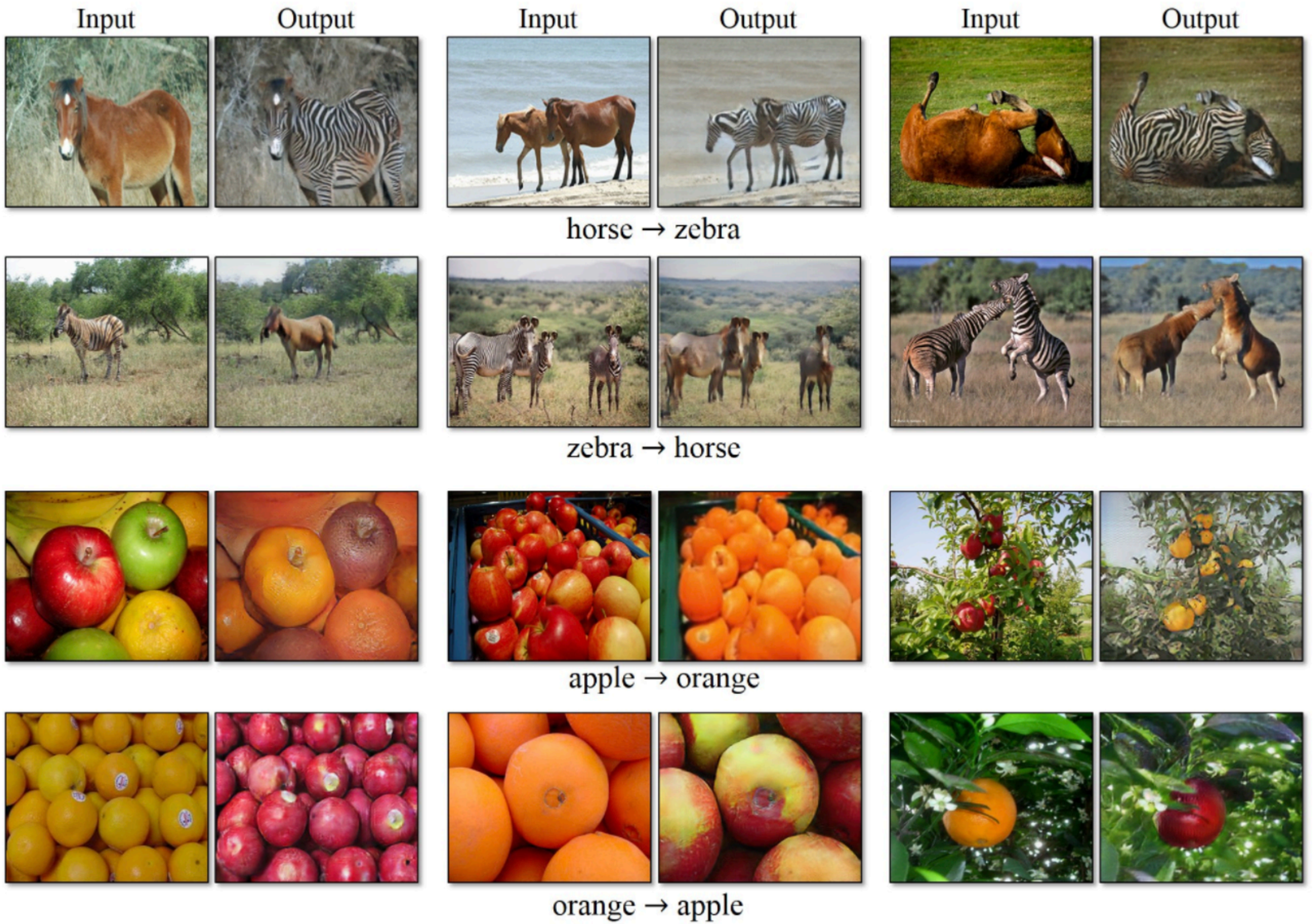
II.D - Interesting results

Super-resolution image:

[Ledig et al. (2016): Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network] Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.D - Interesting results

CycleGANs:



[Jun-Yan Zhu et al. (2017): Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks]

II.D - Interesting results

CycleGANs:

https://hardikbansal.github.io/CycleGANBlog/

Face2ramen

[Jun-Yan Zhu et al. (2017): Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks] Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Announcements

For Tuesday 05/01, 9am:

C2M3

- **Quiz: Hyperparameter tuning, Batch Normalization, Programming Frameworks**
- **Programming assignment: Tensorflow**

C3M1 and C3M2

- **Quiz: Bird recognition in the city of Peacetopia (case study)**
- **Quiz: Autonomous driving (case study)**

For Friday 02/16, 9am:

- **Hands-on session this Friday**

Check out the project example code! (cs230-stanford.github.io**)**

Meet with your mentor (TA), you'll receive a Calendly invite.