Day 9+10: Final Project Presentations & Going Further with ML/DL

Summer STEM: Machine Learning

Nikola Janjušević, Akshaj Kumar Veldanda, Jacky Yuan, Tejaishwarya Gagadam

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

July 18, 2019

Outline

Schedule

1. Finishing Up Projects (9:00-11:00am)
2. Make Presentations (11:00-12:00am)
3. Lunch (12:00pm-1:00pm)
4. Presentations (1:10-3:00pm)
5. Final Material: Going Further with ML/DL (3:00-4:00pm)
6. Goodbye! (4:00pm - ∞?)

Outline

Presentation Guidelines

- 8-10 minutes per group
- Talk about your project:
  - **Goal**: features and targets. What are you predicting?
  - **Dataset**: Where is it from, what does it consist of. Features, targets.
  - **Model**: What is your model? MLP? CNN? What is the architecture? Why?
  - **Training**: How did your model train? Challenges?
  - **Inference**: How was your models predictions on new data?
  - **Going Forward**: "We'd like to continue by..."
  - **Learning**: what did you learn/take away from this project? From this course?
- Put your slides on the master Google-Slides slide-deck (check email)

Outline

1. Schedule
2. Presentation Guidelines
3. Akshaj's Talk
4. Generative Adversarial Networks
5. Course Takeaway

Outline

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before
- How: game between two networks

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before
- How: game between two networks
    - Generator Network

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before
- How: game between two networks
  - Generator Network
  - Discriminator Network

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before
- How: game between two networks
  - Generator Network
  - Discriminator Network
- **Goal of Generator**: generate fake samples indistinguishable from real samples

GANs: Generative Adversarial Networks

- Invented in 2014 by Ian Goodfellow
- Goal: generate samples never seen before
- How: game between two networks
  - Generator Network
  - Discriminator Network
- **Goal of Generator**: generate fake samples indistinguishable from real samples
- **Goal of Discriminator**: be able to tell apart real and fake samples

Implementation

- Generator Model: (on board)

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?
    - Binary Cross Entropy for each

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?
    - Binary Cross Entropy for each
    - Freeze D: Update weights of G to **maximize** BCE

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?
    - Binary Cross Entropy for each
    - Freeze D: Update weights of G to **maximize** BCE
    - Freeze G: Update weights of D to **minimize** BCE

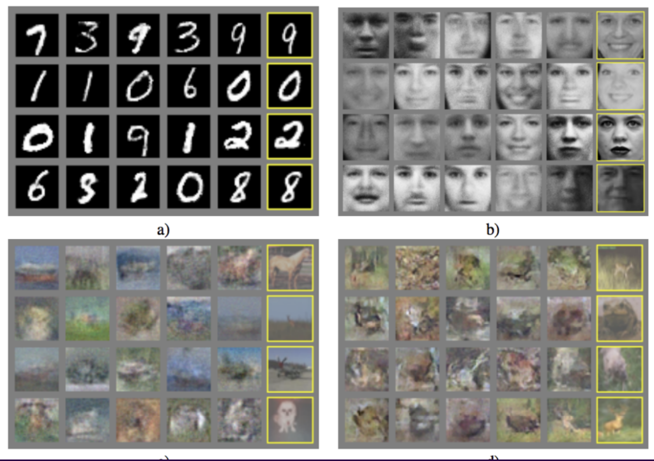
Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?
    - Binary Cross Entropy for each
    - Freeze D: Update weights of G to **maximize** BCE
    - Freeze G: Update weights of D to **minimize** BCE
  - Game-theory: converge to maximin of the cost function

Implementation

- Generator Model: (on board)
  - Draw sample from multi-dim noise vector
  - Up-Convolve into images (learning kernels to generate features)
- Discriminator Model: (on board)
  - Binary Classifier (CNN or MLP)
- Training:
  - Randomly feed Discriminator real and generated samples
  - Balancing act: train D, train G, train D, train G, ...
  - Cost function for each network?
    - Binary Cross Entropy for each
    - Freeze D: Update weights of G to **maximize** BCE
    - Freeze G: Update weights of D to **minimize** BCE
  - Game-theory: converge to maximin of the cost function
- Multi-dim noise vector allows interpolation between samples and can be trained to determine output class of image

Beginning



Generated images (yellow) on a) MNIST b) TFD c) CIFAR-10 (MLP model) d) CIFAR-10 (Conv model)
"Generative Adversarial Networks", Goodfellow et. al. 2014

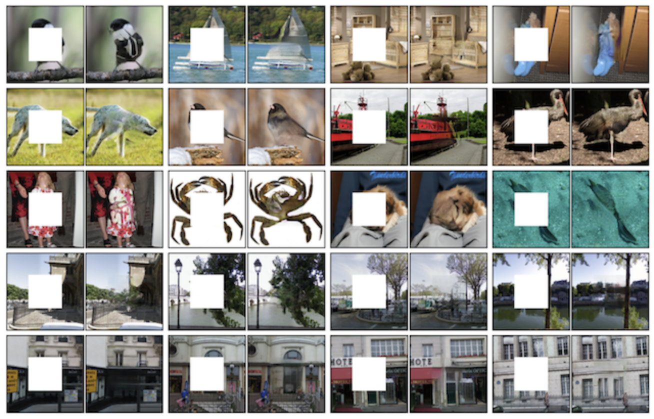
Progress

Improvement of GANs in producing photo-realistic faces over the years

Celebrity Faces

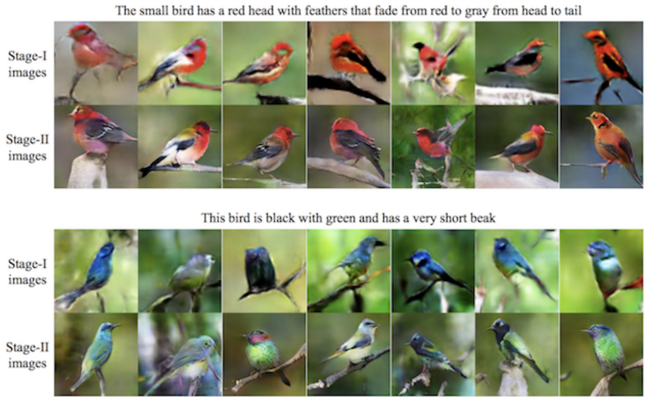
Human face generation one of the most difficult tasks

Occlusion



Recreate missing parts of an image

Text to Image



produce image based on a description

Videos

- Progressive Growing of GANs, 2017
- Few-Shot Adversarial Learning of Realistic Neural Talking Head Models, 2019

Applications of GANs?

- Applications often obscure/tough to think of
- Entering realm of computers outperforming humans on creative tasks

Outline

Course Takeaway

- ML is a lot of Math Math Math Math

Course Takeaway

- ML is a lot of Math Math Math Math
- We've only shown you a subsection

Course Takeaway

- ML is a lot of Math Math Math Math
- We've only shown you a subsection
  - Supervised Learning: Linear/Logistic Regression and Neural Networks

Course Takeaway

- ML is a lot of Math Math Math Math
- We've only shown you a subsection
  - Supervised Learning: Linear/Logistic Regression and Neural Networks
- If you're interested in pursuing further, you know where to look!

Course Takeaway

- ML is a lot of Math Math Math Math
- We've only shown you a subsection
  - Supervised Learning: Linear/Logistic Regression and Neural Networks
- If you're interested in pursuing further, you know where to look!
- If not, that's valuable information for deciding what to do with your future!

Thank You!

- Thank You, Thank You, Thank You!