Data Augmentation through GANs

Eric Gentry
27 Feb 2018
CMPS 292C
Many datasets contain invariances

- Both images contain a dog
  - The “truth” isn’t affected by a reflection
  - But a DNN has to learn that
Many datasets contain invariances

- Both images contain a dog
  - The “truth” isn’t affected by a reflection
  - But a DNN has to learn that

- How to improve it
  - Get more data, or
  - Use data augmentation
Data augmentation teaches invariances

Example Invariance

In this case, the distance between objects might be important, but it \textit{isn't} important that it’s in the lower left.
Example augmentations:
Horizontal reflection

Original

Transformed
Example augmentations: Vertical reflection
Example augmentations: Rotation

Original

Transformed
Example augmentations: Rotation

Original

Transformed

Note: easy to get artifacts if you aren’t careful
Basic Data Augmentation

• Simple transformations:
  • Reflections, translations, zoom, etc.
  • Pre-existing implementation:
    `keras.preprocessing.image.ImageDataGenerator`

• But you can come up with more complicated ones
  • Example: Color balancing

• Requires you to decide which invariances are important
Advanced Data Augmentation

• Can generate simulated images (example paper)
  • Can allow you to “look” at the same objects from different angles

Simulated galaxy images
Advanced Data Augmentation

- Can generate simulated images (example paper)
  - Can allow you to “look” at the same objects from different angles
Advanced Data Augmentation

- Can generate simulated images ([example paper](#))
  - Can allow you to “look” at the same objects from different angles

Simulated

Real

Network fails on images with real, complicated noise
Advanced Data Augmentation

• Can generate simulated images (*example paper*)
  • Can allow you to “look” at the same objects from different angles
  • Can fail on images with real noise unless you:
    • Hand-code a noise model, or
    • Use domain adaptation tricks
Advanced Data Augmentation

- Can generate simulated images
- Example from astronomy
- Can allow you to "look" at the same object from different directions
- Requires you to hand-code noise properties
- Can then finish by training on real images (transfer learning)

![Graph showing R^2 vs. Size training sample](Predicting total brightness)
Problem: requires human input

- A human had to decide:
  - What invariances do we want?
  - What should the noise properties be?
    - Strength of noise?
    - Correlated noise?
  - Is my transformation adding artifacts?

Could a GAN “learn” all this? Maybe.
Using GANs for your augmentation
GANs can learn noise models
GANs can learn noise models

- **Learning from Simulated and Unsupervised Images through Adversarial Training** (Apple FaceID)
  - Humans designed simulated images; GAN learned to make each image “realistic”
GANs can learn noise models

- **Learning from Simulated and Unsupervised Images through Adversarial Training** *(Apple FaceID)*
  - Humans designed simulated images; GAN learned to make each image “realistic”
GANs can learn noise models

• Learning from Simulated and Unsupervised Images through Adversarial Training (Apple FaceID)
  • Humans designed simulated images; GAN learned to make each image “realistic”

• Note: had to tweak loss functions to get best results
  • So still requires some expert intervention
Tweaking required

• Chunked discriminator into local patches
  • Only wanted to get semi-localized noise correct; didn’t want to change global properties of image
• Loss function: summed cross entropy over all patches

Paper: Learning from Simulated and Unsupervised Images through Adversarial Training
Tweaking required

- Chunked discriminator into local patches
  - Only wanted to get semi-localized noise correct; didn’t want to change global properties of image
  - Loss function: summed cross entropy over all patches

Paper: Learning from Simulated and Unsupervised Images through Adversarial Training
Tweaking required

• Chunked discriminator into local patches
• Trained GAN while retaining some examples from previous batches
  • Reduced instability due to GAN re-learning + forgetting to avoid adding image artifacts

Paper: Learning from Simulated and Unsupervised Images through Adversarial Training
Tweaking required

- Chunked discriminator into local patches
- Trained GAN while retaining some examples from previous batches
  - Reduced instability due to GAN re-learning + forgetting to avoid adding image artifacts

Paper: [Learning from Simulated and Unsupervised Images through Adversarial Training](https://example.com)
Comparing Data Augmentation GANs to related GANS
Basic GAN
Conditional GAN

Real images → Gen. → Discr. → Real images

Y → Discr. → Y

z → Gen. → y
Real images

Y

Gen. (frozen)

Z

Disct. (frozen)
Data Augmentation GAN

Real images → Y → Gen. (frozen) → Discr. (frozen) → DNN

z → Gen. (frozen) → Discr. (frozen)
Data Augmentation GAN (multitask)
Attack-generating GAN

(Bo Li mentioned this)
Attack-generating GAN

(Bo Li mentioned this)

So you can mix + match for a variety of problems
Extentions of the Data Augmentation GAN
DAGAN: Extremely versatile

Data Augmentation Generative Adversarial Networks achieved good results for

- Generating samples for a new class given 1 example
- Augmenting standard classifiers in the low-data regime
- Extreme dataset shifts
- ... and more complicated networks

- Uses “UResNet” generator, DenseNet discriminator,
DAGAN: Extremely versatile

Data Augmentation Generative Adversarial Networks needed to:

1. Show they *generated* good samples
2. Show those samples added value to the training process of another network

Paper: [Data Augmentation Generative Adversarial Networks](#)
DAGAN: “UResNet”

Figure 6: UResNet Generator: In this figure one can see a drawing of the UResNet generator as described in Algorithm 1.

Paper: Data Augmentation Generative Adversarial Networks
DAGAN: one-shot generation

Input Image

Autoencoder Gen

U-Net Gen

Res-Net Gen

Previously unseen character (GAN has seen language though)

GANs with the same training sets, but different generator architectures.

Loosely ordered worst to best (left to right)

Paper: Data Augmentation Generative Adversarial Networks
DAGAN: one-shot manifolds

Increasing manifold distance

real

Increasing manifold distance

Real images

Z

Gen.

Paper: Data Augmentation Generative Adversarial Networks
DAGAN: one-shot manifolds

Source domain

Target domain

Paper: Data Augmentation Generative Adversarial Networks
But does DAGAN improve anything?

- Still need to show *classification* is improved

- Test 1: “Vanilla classification”
  - Only 5-15 real examples per class
  - Additional 5x more fake examples
  - **Adds input to classifier indicating if training image is real or fake**

concat(image, real/generated indicator) → DNN

Paper: [Data Augmentation Generative Adversarial Networks](#)
DAGAN always improved classification

<table>
<thead>
<tr>
<th>Omniglot DAGAN Augmented Classification</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Experiment ID</strong></td>
<td><strong>Samples Per Class</strong></td>
<td><strong>Test Accuracy</strong></td>
</tr>
<tr>
<td>Omni_5_Standard</td>
<td>5</td>
<td>0.689904</td>
</tr>
<tr>
<td>Omni_5_DAGAN_Augmented</td>
<td>5</td>
<td><strong>0.821314</strong></td>
</tr>
<tr>
<td>Omni_10_Standard</td>
<td>10</td>
<td>0.794071</td>
</tr>
<tr>
<td>Omni_10_DAGAN_Augmented</td>
<td>10</td>
<td><strong>0.862179</strong></td>
</tr>
<tr>
<td>Omni_15_Standard</td>
<td>15</td>
<td>0.819712</td>
</tr>
<tr>
<td>Omni_15_DAGAN_Augmented</td>
<td>15</td>
<td><strong>0.874199</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMNIST DAGAN Augmented Classification</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Experiment ID</strong></td>
<td><strong>Samples Per Class</strong></td>
<td><strong>Test Accuracy</strong></td>
</tr>
<tr>
<td>EMNIST_Standard</td>
<td>15</td>
<td>0.739353</td>
</tr>
<tr>
<td>EMNIST_DAGAN_Augmented</td>
<td>15</td>
<td><strong>0.760701</strong></td>
</tr>
<tr>
<td>EMNIST_Standard</td>
<td>25</td>
<td>0.783539</td>
</tr>
<tr>
<td>EMNIST_DAGAN_Augmented</td>
<td>25</td>
<td><strong>0.802598</strong></td>
</tr>
<tr>
<td>EMNIST_Standard</td>
<td>50</td>
<td>0.815055</td>
</tr>
<tr>
<td>EMNIST_DAGAN_Augmented</td>
<td>50</td>
<td><strong>0.827832</strong></td>
</tr>
<tr>
<td>EMNIST_Standard</td>
<td>100</td>
<td>0.837787</td>
</tr>
<tr>
<td>EMNIST_DAGAN_Augmented</td>
<td>100</td>
<td><strong>0.848009</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Face DAGAN Augmented Classification</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Experiment ID</strong></td>
<td><strong>Samples Per Class</strong></td>
<td><strong>Test Accuracy</strong></td>
</tr>
<tr>
<td>VGG-Face_Standard</td>
<td>5</td>
<td>0.0446948</td>
</tr>
<tr>
<td>VGG-Face_DAGAN_Augmented</td>
<td>5</td>
<td><strong>0.125969</strong></td>
</tr>
<tr>
<td>VGG-Face_Standard</td>
<td>15</td>
<td>0.39329</td>
</tr>
<tr>
<td>VGG-Face_DAGAN_Augmented</td>
<td>15</td>
<td><strong>0.429385</strong></td>
</tr>
<tr>
<td>VGG-Face_Standard</td>
<td>25</td>
<td>0.579942</td>
</tr>
<tr>
<td>VGG-Face_DAGAN_Augmented</td>
<td>25</td>
<td><strong>0.584666</strong></td>
</tr>
</tbody>
</table>

Paper: [Data Augmentation Generative Adversarial Networks](https://example.com)
But does DAGAN improve anything?

Test 1: Vanilla classification

Paper: Data Augmentation Generative Adversarial Networks
But does DAGAN improve anything?

Test 1: Vanilla classification

Test 2: One-shot learning + Matching Networks
• Matching networks learn a representation space for Nearest Neighbor classification
  • But Nearest Neighbor needs examples

• Has many examples of source domain classes; has only 1 real example per test class
  • Trained DAGAN on source domain, then use sample-selector network to choose 1-2 “best representative” z vectors to augment test class, then do classification

Paper: Data Augmentation Generative Adversarial Networks
DAGAN: one-shot manifolds

Source domain

Target domain

Paper: Data Augmentation Generative Adversarial Networks
DAGAN *sometimes* helps Matching Networks classification

<table>
<thead>
<tr>
<th>Technique Name</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Distance</td>
<td>0.267</td>
</tr>
<tr>
<td>Pixel Distance + DAGAN Augmentations</td>
<td>0.60515</td>
</tr>
<tr>
<td>Matching Nets</td>
<td>0.938</td>
</tr>
<tr>
<td>Neural Statistician</td>
<td>0.931</td>
</tr>
<tr>
<td>Conv. ARC</td>
<td>0.975</td>
</tr>
<tr>
<td>Prototypical Networks</td>
<td>0.96</td>
</tr>
<tr>
<td>Matching Nets (Local Reproduction)</td>
<td>0.969</td>
</tr>
<tr>
<td>Matching Nets + DAGAN Augmentations</td>
<td>0.974</td>
</tr>
</tbody>
</table>

Almost as good as Conv. ARC, which required expert-guided feature engineering

Paper: [Data Augmentation Generative Adversarial Networks](https://example.com)
DAGAN sometimes helps Matching Networks classification

• Worked well for Omniglot, but got mediocre results for EMNIST and VGG-Faces
  • Maybe the Matching Network wasn’t powerful enough?
  • It didn’t hurt the results though
But does DAGAN improve anything?

Test 1: Vanilla classification

Test 2: One-shot learning + Matching Networks
  • Mixed results. Didn’t make it worse, but not always worth the effort.

Paper: Data Augmentation Generative Adversarial Networks
Conclusions

Data Augmentation GANs can help squeeze more information out of your training data.

But there’s a tradeoff:
Rather than spending time hand-coding invariances, you might need to spend time tweaking your GAN.