Big Data EU: Overview of Generative Adversarial Networks (GANs)
About me: Currently helping computer vision & creative startups with their data problems as part of SyntheticLabs.io

- Previously Computer Vision Lead at Founders Factory (Yepic.AI) R&D data scientist at Mudano, previously Data Science Tech Lead at Filtered.com
- Author of Generative Adversarial Networks in Action (2019 / Manning).
- Teaches at University of Birmingham, University of Oxford (AI: Cloud & Edge) and several UK-based financial companies.
- Was in Entrepreneur First 7th cohort (London)
- Graduated from Oxford, also worked there in OUCS.
Today’s talk

1) Briefly touch on (Un)supervised ML
2) Talk about why generation is hard
3) Basis of Generative Adversarial Networks (GANs)
4) Popular architectures:
   a) CycleGAN
   b) BigGAN
5) Adversarial Examples
6) Practical Examples
7) Q & A
Supervised Machine Learning
What is generative modeling? (Unsupervised ML)

- Unsupervised ML and especially generative modeling has been underappreciated until quite recently.
- It is still machine learning—we start with some bad approximation and improve over time.
  - But for that we need to know how well we are doing.
- We are trying to generate examples that look like they came from the original distribution using some learnable function `generate()`.
Generation is hard because the distribution can be complex

- The distribution can be quite complicated
- Evaluation can be difficult too
  - Automatic evaluation is not trivial esp. If we cannot quantify what constitutes a realistic sample
- GANs allow us to implicitly model this complicated distribution, but that also means we don’t get a closed-form solution
What are Generative Adversarial Networks?

@langrjakub
What are the components?

<table>
<thead>
<tr>
<th>Generator</th>
<th>Discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>The Discriminator receives input from two sources:</td>
</tr>
<tr>
<td>A vector of random numbers</td>
<td>● Real examples coming from the training dataset</td>
</tr>
<tr>
<td></td>
<td>● Fake examples coming from the Generator</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>Likelihood that the input example is real</td>
</tr>
<tr>
<td>Fake examples that strive to be as convincing as possible</td>
<td></td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>Distinguish between fake examples coming from the Generator and real examples coming from the training dataset</td>
</tr>
<tr>
<td>Generate fake data that are indistinguishable from members of the training dataset</td>
<td></td>
</tr>
</tbody>
</table>
An analogy: art critic and a forger

Exhibition competition: trying to create the best art exhibit possible.

Source: Dev Nag's blog on medium.com
GANs learn iteratively

@langrjakub
Training objective: Min-Max
(the game theoretical one)

\[ J^D = -0.5E_{x \sim p_d} \log[D(x)] - 0.5E_{z \sim p_g} \log[1 - D(G(z))] \]

\[ J^G = -J^D \]
Training objective: Min-Max training

Figure 1: The contour diagram shows how GAN training is an adversarial process, alternating between minimizing and maximizing the joint value function. The generator $G$ optimizes for orange, and the discriminator $D$ optimizes for purple. If GAN training ends at $(D, G)$, where $G$ is imperfect but $D$ is perfect for that $G$, we can obtain a new generator $G'$ that perfectly models the data distribution by sampling from the $p_D$ distribution.

Figure from Uber Research (https://eng.uber.com/mh-gan/)
Training objective: Non-Saturating
(the one that people actually use)

\[ J^D = -0.5E_{x \sim p_d} \log[D(x)] - 0.5E_{z \sim p_g} \log[1 - D(G(z))] \]

\[ J^G = -0.5E_{z \sim p_g} \log[D(G(z))] \]
So why is this an interesting problem?

- Generative modelling has been largely unsolved. We are moving closer to using *unsupervised learning* as a workable paradigm.
- Variational autoencoders have been the state of the art (best performance) until quite recently.
- Many different domains as well.
Why are GANs so incredible?
CycleGAN: a new approach to domain transfer

- Unpaired domains: cyclical loss
- More complex architecture, but the results are worth it
- Extensions already exist, but e.g. Progressive CycleGAN has not been tried yet

@langrjakub
Idea of cyclical loss. Also remember, that because the loss works both ways, we can now reproduce not just images from summer to winter, but also from winter to summer. In other words if G is our generator from A to B and F is our generator from B to A, then a = F(G(a)) a.
CycleGAN architecture
BigGAN was proposed by DeepMind team in 2018 and presented earlier this year at ICLR.
Generates 512x512 images of all 1,000 classes of ImageNet.
Improvements already exist & people have already started using BigGAN as an upstream preprocessor (e.g. data augmentation).
Artbreeder

Extend your imagination
Adversarial Examples
Adversarial Attacks: Inspiration for GANs

- Adversarial attacks were a predecessor to GANs (2004-ish)
- Adding specially selected noise to the image times some tiny fraction so that for humans it looks the same
- Adversarial Attacks came from the observation that many deep learning models are over-confident even relatively out-of-distribution
Why do adversarial attacks work at all?

- Comes from excessive linearity of the high-dimensional sample space
- Demonstrates the fragility of many existing classifiers, but in practice these tend to be less problematic
- “Looks flashy” and is an “interesting academic problem”
- Many formulations: single pixel attacks, max-$\epsilon$ attacks, region attacks

(image from Ian Goodfellow, 2017)
How bad is it? Does supervised learning suffer from this?

- Modern ML is wrong “almost everywhere”—except for the thin manifold around the training data.
- Gaussian noise (mean and std on the side) through InceptionV3 gives:
  - Ran on a state-of-the-art classifier from 2014
  - Yellow class designation, orange: confidence

@langrjakub
So is ML doomed? — No, let’s think like attackers.

- Generally, this “imperceptible” attacks are quite unrealistic setting
  1. There are simpler attacks if we have that level of access—both time-wise and knowledge wise
  2. Even “well performing” models generally are not perfect in-sample
  3. If we have access to even the model weights, we probably have bigger problems!

But solving this is also hard, essentially requires test error = 0!

Figure 3: Part of assessing the realism of an attack model is determining the economics of how easy an attack is to implement compared to how effective the attack will be. Generating
Practical Applications
Creative Applications: Design at scale

Input (X)  Generated F(X)  Reconstructed G(F(X))

(a) Learning cross-domain relations without any extra label
(b) Handbag images (input) & Generated shoe images (output)
(c) Shoe images (input) & Generated handbag images (output)
Domain Adaptation & Movies

@langrjakub
Despite being hard to train, GANs have delivered many applications previously thought impossible—a real step change in ML research.

Drawing a photorealistic image just from semantic labels is one such example.

There are pretrained models as well as open source implementations.

Image from https://github.com/taki0112/SPADE-Tensorflow
I am not the only one to be excited by this

Example images

Many people (incl. myself) are using these for synthetic data generation.

- If you have severe class imbalance, you can use GANs.
- If you have algorithms that generalize poorly, you can use GANs.
- If you do not have enough data at all, you can use GANs.

(e) The training progression of a successful run.

Figure 4: Note that artifacts appear after around 4.7 million images have been presented to the network. Training recovers shortly after that, however, as can be seen in the diagnostic plots, this failure is not easily detectable from the curves.
More exotic 3D Applications

- Apple, Google have used in production for UX studies, better usability or robotics etc.
- Host of applications that would soon lend themselves to digital asset creation, design, architecture or even engineering.
- So we can learn pretty complex distributions **conditionally**!
- Other interesting purposes include data augmentation beyond simple transforms or style transfer.
CycleGAN: practical applications

Applications Beyond Computer Vision

- Medical Imaging and Biology [Wolterink et al., 20]
- Voice conversion [Fang et al., 2018, Kaneko et al., 2017]
- Cryptography [CipherGAN: Gomez et al., ICLR 2018]
- Robotics
- NLP: Text style transfer.

Deep MR to CT Synthesis using Unpaired Data

Jelmer M. Wolterink¹,², Anna M. Dinkla², Mark H.F. Savenije², Peter R. Seevinck³, Cornelis A.T. van den Berg², Ivana Išgum¹

¹ Image Sciences Institute, University Medical Center Utrecht, The Netherlands j.m.wolterink@umcutrecht.nl
² Department of Radiotherapy, University Medical Center Utrecht, The Netherlands

Input MR  Generated CT  Ground truth CT
Summary: GANs’ applications:
A non-exhaustive lists of academic applications

- Image Synthesis
- Sketch suggestion
- Domain adaptation & transfer
- Style Transfer
- Video style transfer
- Cryptography
- Natural Language Processing
- Video generation
- Text2Image
- Super Resolution
- Image Harmonization
- Text-style transfer
- Compression
- Semi-supervised learning
- Data Augmentation
- 3D model generation
- Data Anonymization
- Audio Synthesis
In context

- “Most interesting idea in ML in the last 10 years” Yann LeCun (Head of AI, Facebook)
- “Pretty much all interesting AI approaches involve GANs in the middle” (Eric Schmidt, 2019 at Stanford HAI)
- GANs have ~2.5x the cited impact of Tensorflow
- Business applications are still young
- As with any new technology, there are still many challenges

@langrjakub
Papers cited


Images

- All animations from the Karras et al. ICLR 2018 presentation are from the official GitHub repository and are under the CC-NC-4.0 license as stated here: https://github.com/tkarras/progressive_growing_of_gans

@langrjakub
Plugs! (... with varying degrees of shame.)

Get the GAN book!

If you want to learn more about using GANs for synthetic data generation or domain adaptation, come talk to me
[jakub@syntheticlabs.io](jakub@syntheticlabs.io)
Thank you!

Any questions?

🌐 jakublangr.com
🐦 @langrjakub