Deep Generative Models

Greg Mori
School of Computing Science
Simon Fraser University
Outline

• Introduction to Variational Auto-Encoders (VAEs) and Generative Adversarial Networks (GANs)

• Examples of VAEs for generation
  Video generation (He et al. ECCV 2018)
  Scene generation (Deng et al. NeurIPS 2018)

• Limitations of deep generative models
Figure 5: 1024×1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.


Figure 6: Visual quality comparison in LSUN BEDROOM; pictures copied from the cited articles.

Our contributions allow us to deal with high output resolutions in a robust and efficient fashion. Figure 5 shows selected 1024×1024 images produced by our network. While megapixel GAN results have been shown before in another dataset (Marchesi, 2017), our results are vastly more varied and of higher perceptual quality. Please refer to Appendix F for a larger set of result images as well as the nearest neighbors found from the training data. The accompanying video shows latent space interpolations and visualizes the progressive training. The interpolation works so that we first randomize a latent code for each frame (512 components sampled individually from $N(0, 1)$), then blur the latents across time with a Gaussian ($= 45$ frames @ 60Hz), and finally normalize each vector to lie on a hypersphere.

We trained the network on 8 Tesla V100 GPUs for 4 days, after which we no longer observed qualitative differences between the results of consecutive training iterations. Our implementation used an adaptive minibatch size depending on the current output resolution so that the available memory budget was optimally utilized.

In order to demonstrate that our contributions are largely orthogonal to the choice of a loss function, we also trained the same network using LSGAN loss instead of WGAN-GP loss. Figure 1 shows six examples of 1024×2 images produced using our method using LSGAN. Further details of this setup are given in Appendix B.
Generative Adversarial Networks

Sample a random code - Represents properties of the thing to generate

$\mathbf{z} \sim p(\mathbf{z})$

Generator maps a random code to an output - Generator is a (deep, convolutional) neural network

$G(\mathbf{z}, \theta_g)$

$x$
Generative Adversarial Networks

How do we learn the parameters of the generator? No labeled data? But lots of real unlabeled data…
But which face should we generate?
What is a good output?
What if we generate non-faces?
Overfitting?
Variety?
Given a discriminator, optimize the generator to fool the discriminator
- Generator should generate images that the discriminator thinks are real images, minimize wrt $G$:

$$\mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z, \theta_g), \theta_d))]$$
Real or Fake prob.

How do we train the discriminator?
\[ \mathbf{x} \sim p_{\text{data}}(\mathbf{x}) \]

\[ D(\mathbf{x}, \theta_d) \]

Over the real images, make sure the discriminator thinks they are real
Maximize wrt. D:

\[ \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[ \log D(\mathbf{x}, \theta_d) \right] \]
\[
\min_{\theta_g} \max_{\theta_d} \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x, \theta_d)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z, \theta_g), \theta_d))]
\]
Real or Fake prob.

Alternating minimization:
- Hold G fixed, max over D
- Hold D fixed, min over G
- Rinse and repeat

$$\min_{\theta_g} \max_{\theta_d} \mathbb{E}_{x \sim p_{data}(x)}[\log D(x, \theta_d)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z, \theta_g), \theta_d))]$$
Alternating minimization:
- Hold G fixed, max over D
- Hold D fixed, min over G
- Rinse and repeat

Once finished, discard the discriminator
Generator should generate realistic images for samples of $z$
Variational Autoencoders

\[ z \sim p(z) \]

Sample a random code
- Represents properties of the thing to generate

\[ p_{\theta}(x|z) \]

Decoder maps a random code to a distribution over outputs
- Decoder is a (deep, convolutional) neural network

\[ x \sim p_{\theta}(x|z) \]
Train to maximize likelihood

\[ \theta = \operatorname{arg \, max}_\theta p(x) \]

\[ p(x) = \int p(x, z)dz \]

\[ = \int p_\theta(x|z)p(z)dz \]
Train to maximize likelihood

\[
p(x) \approx \frac{1}{N} \sum_{i=1}^{N} p_\theta(x | z^{(i)})
\]

Sampling is a problem, many z samples with low probability
Encoder maps an input to a distribution over codes
- Encoder is a (deep, convolutional) neural network

Encoder maps $x$ to distribution over $z$ with high probability of generating $x$
• But we were supposed to do

\[ p(x) \approx \frac{1}{N} \sum_{i=1}^{N} p_{\theta}(x|z^{(i)}) , \quad z^{(i)} \sim p(z) \]
• Let’s start by looking at the difference between the new “proposal” distribution $q$ we are using and the “true posterior” $p$.

$$KL(q(z|x)||p(z|x)) \equiv \mathbb{E}_{z \sim q}[\log q(z|x) - \log p(z|x)]$$
• Rearranging

\[ \log p(x) - KL(q(z|x)\|p(z|x)) = \mathbb{E}_{z \sim q}[\log p(x|z)] - KL(q(z|x)\|p(z)) \]

Positive: how different is approximate posterior from tru?

How well can samples from approximate posterior generate the data?

Regularize approximate posterior by prior
\[ p(\mathbf{z}) \]

\[ KL(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \]

\[ q_\phi(\mathbf{z}|\mathbf{x}) \]

\[ p_\theta(\mathbf{x}|\mathbf{z}) \]
\[
\log p(x) \geq \mathbb{E}_{z \sim q_\phi} \left[ \log p(x|z) \right] + KL(q_\phi(z|x) \| p(z))
\]
Outline

• Introduction to Variational Auto-Encoders (VAEs) and Generative Adversarial Networks (GANs)

• Examples of VAEs for generation
  Video generation (He et al. ECCV 2018)
  Scene generation (Deng et al. NeurIPS 2018)

• Limitations of deep generative models
Model

• Non-parametric density estimation
• Maximum a posteriori conditional likelihood

• ICCV 2003
• Data inefficient
• Weak generalization

Copy pixels

2003

Generate pixels (GANs, VAEs)

2018

• Data efficient
• Compositionality
• Modularity
Probabilistic Video Generation using Holistic Attribute Control
Video Generation and Forecasting

**Video Generation:** Sample should look like a real video

(REAL results from our model)

**Video Forecasting:** Conditioned on the first frame, future should be realistic

(REAL results from our model)
Challenges

Appearance has to be visually realistic

Futures are non-deterministic, so the model needs to capture multi-modal distributions

[ Vondrick et al., NIPS 2016 ]
Variational Autoencoder (VAE)
Variational Autoencoder (VAE) + LSTM
Challenges

Appearance has to be visually realistic

Futures are non-deterministic, so the model needs to capture multi-modal distributions

There needs to be temporal consistency (e.g., identity should not change), which means that it must maintain mode consistency

It would nice to have some high level control over generated videos
VAE + LSTM with Conditional Sampling
VAE + LSTM with Structured Latent Space
Holistic Attribute Control

\[ \psi(t) \]

LSTM

Prior

Dynamic Approximate Posterior

Conditional Approximate Posterior

Residual Appearance

Initial Approximate Posterior

Controlled Appearance

\( a_1, a_2, a_n \)

Holistic Attribute Control

\( \phi_{\text{att}} \)
Results: Chair CAD dataset

Ablation

<table>
<thead>
<tr>
<th>Bound</th>
<th>Static</th>
<th>$-C$</th>
<th>$-S$</th>
<th>$+S$</th>
<th>$+C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-E ↓</td>
<td>1.98</td>
<td>40.33</td>
<td>17.64</td>
<td>7.79</td>
<td>14.81</td>
</tr>
<tr>
<td>Inter-E ↑</td>
<td>1.39</td>
<td>0.42</td>
<td>0.73</td>
<td>1.35</td>
<td>1.02</td>
</tr>
<tr>
<td>I-Score ↑</td>
<td>4.01</td>
<td>1.28</td>
<td>1.83</td>
<td>3.63</td>
<td>2.56</td>
</tr>
</tbody>
</table>

Quantitative

| Chair CAD [1, 40] |
|-------------------|-----------------|----------------|
| Bound | Deep Rot. [40] | VideoVAE (ours) |
| Intra-E ↓ | 1.98 | 14.68 | 5.50 |
| Inter-E ↑ | 1.39 | 1.34 | 1.37 |
| I-Score ↑ | 4.01 | 3.39 | 3.94 |
**Results:** Weizmann Human Action dataset

- **Identity** = ◆ | ▲ | ●
- **Action** = ○ walking | ● running | ● skipping | ● jumping jack | ● side step

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-E</td>
<td>↓ 0.63</td>
<td>23.58</td>
<td>9.53</td>
</tr>
<tr>
<td>Inter-E</td>
<td>↑ 4.49</td>
<td>2.91</td>
<td>4.37</td>
</tr>
<tr>
<td>I-Score</td>
<td>↑ 89.12</td>
<td>13.87</td>
<td>69.55</td>
</tr>
</tbody>
</table>
## Results: MIT Flickr

![MIT Flickr Images](image_url)

<table>
<thead>
<tr>
<th>YFCC [31] — MIT Flickr [34]</th>
<th>Bound</th>
<th>VGAN [34]</th>
<th>VideoVAE (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-E ▼ 30.34</td>
<td>46.96</td>
<td>44.03</td>
<td><strong>38.20</strong></td>
</tr>
<tr>
<td>Inter-E ▲ 0.693</td>
<td><strong>0.692</strong></td>
<td>0.691</td>
<td>0.692</td>
</tr>
<tr>
<td>I-Score ▲ 1.87</td>
<td>1.58</td>
<td>1.62</td>
<td><strong>1.81</strong></td>
</tr>
</tbody>
</table>
Probabilistic Neural Programmed Networks for Scene Generation

Zhiwei Deng
Jiacheng Chen
Yifang Fu
Greg Mori

NIPS 2018
Scene generation problem

A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.
Scene generation problem

A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.

Attributes
A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.

**Objects**
Scene generation problem

A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.

*Relations*
Scene generation problem

A blue metal cube is behind a cyan rubber sphere with a red metal sphere next to it.

Challenges:
- description variability
- scene variability
- compositionality
- learning efficiency
Our Approach: Probabilistic Neural Programmed Networks
Existing Approaches

\[ y = \text{“a shiny green cube next to a shiny silver cylinder with a matte red cube in front and a rubber cyan cube behind”} \]

\[ \text{z from prior + condition } y \] \quad \text{encode condition } y

GAN, VAE

\[ \text{e.g. DCGAN} \]

Monolithic

Autoregressive

\[ \text{e.g. PixelCNN, PixelVAE} \]

Fine details, but lacks structure
Our Proposed Model

\[ y = \text{“a red metal sphere next to a cyan rubber sphere with a blue metal cube behind”} \]
Our Proposed Model

\[ y = \text{“a red metal sphere next to a cyan rubber sphere with a blue metal cube behind”} \]
Our Proposed Model

\[ y = \text{"a red metal sphere next to a cyan rubber sphere with a blue metal cube behind" } \]
Concept mapping operator
• E.g. sphere, cube, cylinder

Concept one hot encoding

\[ [0, ..., 1, 0, ..., 0] \]

\[ N(\mu_a, \sigma_a) \quad N(\mu_s, \sigma_s) \]
Reusable Operators

Combine operator (compound attribute)
• E.g. red shiny, blue matte

PoE: Product of Experts
Reusable Operators

Describe operator (object-dependent combination)
• E.g. red shiny sphere, blue matte cube
Reusable Operators

Transform operator (Instantiate an appearance distribution)
- E.g. red shiny sphere with size 15x18 pixels
Reusable Operators

Layout operator (arrange positions for objects)
- E.g. red shiny sphere at location 41,28 with blue matte cube at 19,31
Applying Modules

... red sphere
... Next to
... cyan cube

Scene distribution

Decoder
Learning Process

\[ p(z \mid y) \]

\[ KL(q \parallel p) \]

\[ q(z \mid x) \]
Experiments: Performance Measure

Evaluation: Detector score
Experiments: Performance Measure

**Evaluation:** Detector score

Faster-RCNN

Mean average precision (mAP)

Green metal sphere
Purple metal cylinder
Purple metal sphere

…
Experiments: Comparisons

**Baselines:**

- LSTM+DCGAN
- LSTM+PixelCNN
- Product of experts + VAE
Experiments: Datasets

Color MNIST
8000 training, 8000 test

CLEVR-G
10000 training, 10000 test
## Experiments: Results

### Color MNIST

<table>
<thead>
<tr>
<th></th>
<th>GT</th>
<th>Ours</th>
<th>LSTM+ DCGAN</th>
<th>LSTM+ PixelCNN</th>
<th>PoE+ VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectness</strong></td>
<td>99.9</td>
<td>98.1</td>
<td>99.0</td>
<td>92.1</td>
<td>85.8</td>
</tr>
<tr>
<td><strong>Object type correctness</strong></td>
<td>99.9</td>
<td>41.9</td>
<td>21.1</td>
<td>47.4</td>
<td>12.6</td>
</tr>
<tr>
<td><strong>Object attribute correctness</strong></td>
<td>99.9</td>
<td>36.3</td>
<td>14.6</td>
<td>31.8</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Experiments: Results

Color MNIST (generated samples)

cyan 7
left

yellow 2

green 8
top

red 1
## Experiments: 4-object Scenes

**CLEVR-G (up to 4 objects)**

<table>
<thead>
<tr>
<th></th>
<th>GT</th>
<th>Ours</th>
<th>LSTM+ DCGAN</th>
<th>LSTM+ PixelCNN</th>
<th>PoE+ VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objectness</strong></td>
<td>99.9</td>
<td>97.1</td>
<td>97.9</td>
<td>89.4</td>
<td>97.4</td>
</tr>
<tr>
<td><strong>Object type correctness</strong></td>
<td>99.8</td>
<td>83.3</td>
<td>56.6</td>
<td>44.4</td>
<td>49.3</td>
</tr>
<tr>
<td><strong>Object attribute correctness</strong></td>
<td>97.6</td>
<td>73.7</td>
<td>17.6</td>
<td>7.4</td>
<td>13.4</td>
</tr>
</tbody>
</table>
## Experiments: Unseen Object-Attribute Combinations

**CLEVR-G (up to 4 objects, zero-shot combination)**

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>LSTM+ DCGAN</th>
<th>LSTM+ PixelCNN</th>
<th>PoE+ VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectness</td>
<td>97.0</td>
<td>98.9</td>
<td>89.9</td>
<td>97.5</td>
</tr>
<tr>
<td>Object type correctness</td>
<td>75.2</td>
<td>41.8</td>
<td>42.0</td>
<td>44.1</td>
</tr>
<tr>
<td>Object attribute correctness</td>
<td>73.4</td>
<td>33.2</td>
<td>19.2</td>
<td>31.8</td>
</tr>
</tbody>
</table>
### Experiments: 8-object Scenes

**CLEVR-G (train on up to 8 objects)**

<table>
<thead>
<tr>
<th></th>
<th>Objectness</th>
<th>Object type correctness</th>
<th>Object attribute correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT</strong></td>
<td>97.9</td>
<td>97.7</td>
<td>94.3</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>97.3</td>
<td>72.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>
**Experiments: Generalize to More Objects**

**CLEVR-G (train on up to 4 objects, test on up to 8 objects)**

<table>
<thead>
<tr>
<th></th>
<th>Objectness</th>
<th>Object type correctness</th>
<th>Object attribute correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT</td>
<td>97.9</td>
<td>97.7</td>
<td>94.3</td>
</tr>
<tr>
<td>Ours</td>
<td>97.6</td>
<td>71.5</td>
<td>51.8</td>
</tr>
</tbody>
</table>
## Experiments: Qualitative Results

<table>
<thead>
<tr>
<th></th>
<th>GT</th>
<th>PNP Net</th>
<th>DCGAN</th>
<th>PixelCNN</th>
<th>GT(ZS)</th>
<th>PNP Net (ZS)</th>
<th>DCGAN (ZS)</th>
<th>PixelCNN (ZS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 object</strong></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>2 objects</strong></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
<tr>
<td><strong>4 objects</strong></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Experiments: Variety of Scenes
Outline

• Introduction to Variational Auto-Encoders (VAEs) and Generative Adversarial Networks (GANs)

• Examples of VAEs for generation
  Video generation (He et al. ECCV 2018)
  Scene generation (Deng et al. NeurIPS 2018)

• Limitations of deep generative models
Mode Collapse

• Particular problem for GANs
  • GAN generator learns to produce a small set of different examples
  • Covers one or a few modes in the distribution

• If the goals is to fool the discriminator, do so by being very good at a particular example
  • Jump around to keep the discriminator on its toes
  • Interesting relation between max-min, min-max, and gradients

• Much research focused on how to reduce mode collapse in GANs
Real or Fake?
Real or Fake?

Bau et al. ICCV 2019
LSUN Church images
Reading list

- J. He, A. Lehrmann, J. Marino, L. Sigal, G. Mori. Probabilistic Video Generation using Holistic Attribute Control. ECCV 2018


- I. Goodfellow, Y. Bengio and A. Courville. Deep Learning (Ch. 20)