PixelGAN Autoencoders

Alireza Makhzani, Brendan Frey

Machine learning Group
University of Toronto

CIFAR Deep Learning Summer School
Montreal, Canada

June 29th, 2017
1. Background
   - PixelCNNs
   - Variational Autoencoders
   - Adversarial Autoencoders

2. PixelGAN Autoencoders
   - Gaussian Priors
   - Categorical Priors
     ✦ Clustering
     ✦ Semi-supervised Learning
PixelCNNs

- Learn the image statistics directly at the pixel level.
- Good at modelling low-level pixel statistics.
- Samples lack global structure.
- Lacking latent representation.
- Conditional PixelCNNs can learn conditional densities.

van den Oord et al., 2016
Variational Autoencoders

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

- Good at capturing the global structure, but samples are blurry.
- Learn hierarchical representations useful for downstream tasks.
- Attempts at combining PixelCNN with VAEs:
  - PixelVAE (Gulrajani et al., 2016)
  - VLAE (Chen et al., 2017)

Kingma et al., 2013
Adversarial Autoencoders

\[ q(z) = \int_q q(z|x)p_{data}(x)dx \]

Code Space of MNIST:

\[ \text{Gaussian Prior} \quad \text{Mixture of Gaussians} \]

Makhzani et al., 2015
Outline

1. Background
   - PixelCNNs
   - Variational Autoencoders
   - Adversarial Autoencoders

2. PixelGAN Autoencoders
   - Gaussian Priors
   - Categorical Priors
     ✦ Clustering
     ✦ Semi-supervised Learning
All the image statistics are captured by the single latent vector.

\[ p(z) \]

**Latent Variable**

\[ p(x|z) \]

**Deterministic**

(factorized Gaussians)

**VAE**

- label, style
- global and local

None
The image statistics are captured jointly by the latent vector and the autoregressive decoder.

\[ p(z) \]

\[ p(x|z) \]

Latent Variable

PixelCNN
The image statistics are captured jointly by the latent vector and the autoregressive decoder.
The image statistics are captured jointly by the latent vector and the autoregressive decoder.
The image statistics are captured jointly by the latent vector and the autoregressive decoder.
Cost function of PixelGAN = Reconstruction + Adversarial Cost
Outline

1. Background
   - PixelCNNs
   - Variational Autoencoders
   - Adversarial Autoencoders

2. PixelGAN Autoencoders
   - Gaussian Priors
   - Categorical Priors
   - Clustering
   - Semi-supervised Learning
Global vs. Local Decomposition

(a) PixelGAN Samples (2D code, limited receptive field)
(b) PixelCNN Samples (limited receptive field)
(c) AAE Samples (2D code)
Code Space

Code Space of MNIST:
Outline

1. Background
   - PixelCNNs
   - Variational Autoencoders
   - Adversarial Autoencoders

2. PixelGAN Autoencoders
   - Gaussian Priors
   - Categorical Priors
     - Clustering
     - Semi-supervised Learning
Figure 4: Architecture of the PixelGAN autoencoder with the categorical prior. $p(z)$ captures the class label and $p(x|z)$ is a multi-modal distribution that captures the style distribution of a digit conditioned on the class label of that digit.
Discrete vs. Continuous Decomposition (Clustering)

Figure 5: Effect of GAN regularization on the code space of PixelGAN autoencoders: (a) no distribution is imposed on the hidden code. (b) a categorical prior is imposed on the hidden code.
Discrete vs. Continuous Decomposition (Clustering)

0.3% Error rate

![Diagram of PixelGAN Autoencoders](image)

Figure 5: Effect of GAN regularization on the code space of PixelGAN autoencoders: (a) no distribution is imposed on the hidden code. (b) a categorical prior is imposed on the hidden code.
Unsupervised Clustering

PixelGAN Autoencoders
Figure 6: Disentangling the content and style in an unsupervised fashion with PixelGAN autoencoders. Each row shows samples of the model from one of the learnt clusters.
Figure 6: Disentangling the content and style in an unsupervised fashion with PixelGAN autoencoders. Each row shows samples of the model from one of the learnt clusters.
Outline

1. Background
   - PixelCNNs
   - Variational Autoencoders
   - Adversarial Autoencoders

2. PixelGAN Autoencoders
   - Gaussian Priors
   - Categorical Priors
     ✦ Clustering
     ✦ Semi-supervised Learning
Semi-supervised Learning
Semi-supervised Learning

(a) SVHN (1000 labels)
(b) MNIST (100 labels)
(c) NORB (1000 labels)

Figure 8: Conditional samples of the semi-supervised PixelGAN autoencoder.
Semi-supervised Classification

Figure 8: Semi-supervised error-rate of PixelGAN autoencoders on the MNIST and SVHN datasets.

<table>
<thead>
<tr>
<th></th>
<th>MNIST (Unsupervised)</th>
<th>MNIST (20 labels)</th>
<th>MNIST (50 labels)</th>
<th>MNIST (100 labels)</th>
<th>SVHN (500 labels)</th>
<th>SVHN (1000 labels)</th>
<th>NORB (1000 labels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE [25]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.33 (±0.14)</td>
<td>-</td>
<td>36.02 (±0.10)</td>
<td>18.79 (±0.05)</td>
</tr>
<tr>
<td>VAT [26]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.33</td>
<td>-</td>
<td>24.63</td>
<td>9.88</td>
</tr>
<tr>
<td>ADGM [27]</td>
<td>-</td>
<td>-</td>
<td>0.96 (±0.02)</td>
<td>-</td>
<td>-</td>
<td>22.86</td>
<td>10.06 (±0.05)</td>
</tr>
<tr>
<td>SDGM [27]</td>
<td>-</td>
<td>-</td>
<td>1.32 (±0.07)</td>
<td>-</td>
<td>-</td>
<td>16.61 (±0.24)</td>
<td>9.40 (±0.04)</td>
</tr>
<tr>
<td>Adversarial Autoencoder [6]</td>
<td>4.10 (±1.13)</td>
<td>-</td>
<td>-</td>
<td>1.90 (±0.10)</td>
<td>-</td>
<td>17.70 (±0.30)</td>
<td>-</td>
</tr>
<tr>
<td>Ladder Networks [28]</td>
<td>-</td>
<td>-</td>
<td>0.89 (±0.50)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Convolutional CatGAN [24]</td>
<td>4.27</td>
<td>-</td>
<td>-</td>
<td>1.39 (±0.28)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>InfoGAN [18]</td>
<td>-</td>
<td>5.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Feature Matching GAN [29]</td>
<td>-</td>
<td>16.77 (±4.52)</td>
<td>2.21 (±1.36)</td>
<td>0.93 (±0.06)</td>
<td>18.44 (±4.80)</td>
<td>8.11 (±1.30)</td>
<td>-</td>
</tr>
<tr>
<td>Temporal Ensembling [30]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.05 (±0.30)</td>
<td>5.43 (±0.25)</td>
<td>-</td>
</tr>
<tr>
<td><strong>PixelGAN Autoencoders</strong></td>
<td><strong>5.27 (±1.81)</strong></td>
<td><strong>12.08 (±5.50)</strong></td>
<td><strong>1.16 (±0.17)</strong></td>
<td><strong>1.08 (±0.15)</strong></td>
<td><strong>10.47 (±1.80)</strong></td>
<td><strong>6.96 (±0.55)</strong></td>
<td><strong>8.90 (±1.0)</strong></td>
</tr>
</tbody>
</table>

Table 1: Semi-supervised learning and clustering error-rate on MNIST, SVHN and NORB datasets.
Thank you!