Bidirectional GAN

Adversarially Learned Inference (ICLR 2017)
V. Dumoulin\textsuperscript{1}, I. Belghazi\textsuperscript{1}, B. Poole\textsuperscript{2}, O. Mastropietro\textsuperscript{1}, A. Lamb\textsuperscript{1}, M. Arjovsky\textsuperscript{3} and A. Courville\textsuperscript{1}
\textsuperscript{1}Universite de Montreal & \textsuperscript{2}Stanford University & \textsuperscript{3} New York University

Adversarial Feature Learning (ICLR 2017)
J. Donahue, P. Krahenbuhl and T. Darrell
University of California, Berkeley

Presented by Hao Liu
June 9, 2017
GAN: Main Idea

- An adversarial game between:
  1. Discriminator D
  2. Generator G

- Discriminator is trained to discriminate between samples $x$ from:
  1. Real data distribution $q(x)$
  2. Generated data distribution $p(x) = \int p(z)p(x|z)dz$, implemented via a generator by sampling $z \sim p(z)$ and then sampling $x \sim p(x|z)$

- Generator learns conditional $p(x|z)$ to fool the discriminator
GAN: Formulation

- The adversarial game is formalized by the value function

\[
\min_G \max_D V(D, G) \quad (1)
\]
\[
= \mathbb{E}_{q(x)}[\log(D(x))] + \mathbb{E}_{p(z)}[\log(1 - D(G(z)))] \quad (2)
\]
\[
= \int q(x) \log(D(x))dx + \int\int p(z)p(x|z)\log(1 - D(x))dx dz \quad (3)
\]

- Lacks an efficient mechanism to infer \( z \) given \( x \)
Bidirectional GAN: Main Idea

- Cast the learning of both an inference machine (encoder) and a deep directed generative model (decoder) in a GAN-like adversarial framework.
- Discriminator is trained to discriminate between joint samples \((x,z)\) from:
  1. Encoder distribution \(q(x,z) = q(x)q(z|x)\), or
  2. Decoder distribution \(p(x,z) = p(z)p(x|z)\)
- Generator learns conditionals \(q(z|x)\) and \(p(x|z)\) to fool the discriminator
Bidirectional GAN: Formulation

The adversarial game is formalized by the value function

\[
\min_G \max_D V(D, G) \quad \text{(4)}
\]

\[
= \mathbb{E}_{q(x)}[\log(D(x, G_z(x)))] + \mathbb{E}_{p(z)}[\log(1 - D(G_x(z), z))] \quad \text{(5)}
\]

\[
= \int \int q(x)q(z | x) \log(D(x, z))dxdz + \int \int p(z)p(x | z) \log(1 - D(x, z))dxdz \quad \text{(6)}
\]
Algorithm 1 The ALI training procedure.

\[ \theta_g, \theta_d \leftarrow \text{initialize network parameters} \]

repeat
\[ \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)} \sim q(x) \]
\[ \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(M)} \sim p(z) \]
\[ \hat{z}^{(i)} \sim q(z | \mathbf{x} = \mathbf{x}^{(i)}), \quad i = 1, \ldots, M \]
\[ \tilde{x}^{(j)} \sim p(x | z = \mathbf{z}^{(j)}), \quad j = 1, \ldots, M \]
\[ \rho_q^{(i)} \leftarrow D(\mathbf{x}^{(i)}, \hat{z}^{(i)}), \quad i = 1, \ldots, M \]
\[ \rho_p^{(j)} \leftarrow D(\tilde{x}^{(j)}, z^{(j)}), \quad j = 1, \ldots, M \]
\[ \mathcal{L}_d \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(\rho_q^{(i)}) - \frac{1}{M} \sum_{j=1}^{M} \log(1 - \rho_p^{(j)}) \]
\[ \mathcal{L}_g \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(1 - \rho_q^{(i)}) - \frac{1}{M} \sum_{j=1}^{M} \log(\rho_p^{(j)}) \]
\[ \theta_d \leftarrow \theta_d - \nabla_{\theta_d} \mathcal{L}_d \]
\[ \theta_g \leftarrow \theta_g - \nabla_{\theta_g} \mathcal{L}_g \]
until convergence

Notes:

- BiGAN uses two deterministic mappings
- ALI’s experiments use one deterministic mapping and one stochastic mapping
Bidirectional GAN: Theoretical Properties

**Theorem (optimal discriminator and generator)**

- For a fixed generator, the optimal discriminator is
  \[ D^*(x, z) = \frac{q(x,z)}{q(x,z)+p(x,z)} \]
- The encoder and decoder’s objective for an optimal discriminator can be rewritten as
  \[ C(G) \triangleq \max_D V(D, G) = V(D^*, G) = 2D_{JS}(p(x, z) \| q(x, z)) - \log 4 \]
- The global minimum of \( C(G) \) is achieved if and only if
  \[ p(x, z) = q(x, z) \]. At that point, \( C(G) = -\log 4 \) and \( D^*(x, z) = \frac{1}{2} \)

**Theorem (reconstruction)**

- Assuming optimal discriminator \( D \) and generator \( G \). If the encoder \( G_z \) and decoder \( G_x \) are deterministic, then
  \[ G_x = G_z^{-1} \text{ and } G_z = G_x^{-1} \]
  almost everywhere
ALI Experiments: Samples and Reconstruction

- can hardly reconstruct because of underfitting in practice

**Figure:** For the reconstructions, odd columns are original samples from the validation set and even columns are corresponding reconstructions.
ALI Experiments: Semi-supervised Learning

<table>
<thead>
<tr>
<th>Model</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
<th>8000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ladder network (Rasmus et al., 2015)</td>
<td></td>
<td></td>
<td>20.40</td>
<td></td>
</tr>
<tr>
<td>CatGAN (Springenberg, 2015)</td>
<td></td>
<td></td>
<td>19.58</td>
<td></td>
</tr>
<tr>
<td>GAN (feature matching) (Salimans et al., 2016)</td>
<td>21.83 ± 2.01</td>
<td>19.61 ± 2.09</td>
<td>18.63 ± 2.32</td>
<td>17.72 ± 1.82</td>
</tr>
<tr>
<td>ALI (ours, no feature matching)</td>
<td>19.98 ± 0.89</td>
<td>19.09 ± 0.44</td>
<td>17.99 ± 1.62</td>
<td>17.05 ± 1.49</td>
</tr>
</tbody>
</table>

**Figure:** CIFAR10 test set missclassification rate for semi-supervised learning using different numbers of trained labeled examples

- The discriminator \( D \) takes \( x \) and \( z \) as input and outputs a distribution over \( K + 1 \) classes, where \( K \) is the number of categories.
- For labeled \( q(x, z) \) samples, \( D \) is expected to predict the label.
- For unlabeled \( q(x, z) \) and \( p(x, z) \) samples, \( D \) is expected to predict \( K + 1 \) for \( p(x, z) \) and \( k \in \{1, \ldots, K\} \) for \( q(x, z) \) samples.
BiGAN Experiments: MNIST

<table>
<thead>
<tr>
<th></th>
<th>BiGAN</th>
<th>D</th>
<th>LR</th>
<th>JLR</th>
<th>AE ($l_2$)</th>
<th>AE ($l_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>97.39</td>
<td>97.30</td>
<td>97.44</td>
<td>97.13</td>
<td>97.58</td>
<td>97.63</td>
</tr>
</tbody>
</table>

**Figure:** 1NN classification accuracy on permutation-invariant MNIST test set in the feature space learned by BiGAN, Latent Regressor (LR), Joint Latent Regressor (JLR), and an autoencoder (AE) using an L1 or L2 distance.

- **LR:** Latent Regressor, first train a GAN, then learn an encoder to reconstruct $z$
- **JLR:** Joint Latent Regressor, jointly learn an encoder to reconstruct $z$

$G(z)\begin{array}{c}73614816186630210467\end{array}$

$\begin{array}{c}x\end{array}\begin{array}{c}01234567890123456789\end{array}$

$G(E(x))\begin{array}{c}01237507013444757\end{array}$

**Figure:** generator samples $G(z)$, real data $x$, and corresponding reconstructions $G(E(x))$ on the permutation-invariant MNIST
Figure: generator samples $G(z)$, real data $x$, and corresponding reconstructions $G(E(x))$ on ImageNet
BiGAN Experiments: Transferability

Figure: Classification accuracy for the ImageNet LSVRC validation set with various portions of the network frozen, or reinitialized and trained from scratch

<table>
<thead>
<tr>
<th>Method</th>
<th>conv1</th>
<th>conv2</th>
<th>conv3</th>
<th>conv4</th>
<th>conv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random (Noroozi &amp; Favaro, 2016)</td>
<td>48.5</td>
<td>41.0</td>
<td>34.8</td>
<td>27.1</td>
<td>12.0</td>
</tr>
<tr>
<td>Wang &amp; Gupta (2015)</td>
<td>51.8</td>
<td>46.9</td>
<td>42.8</td>
<td>38.8</td>
<td>29.8</td>
</tr>
<tr>
<td>Doersch et al. (2015)</td>
<td>53.1</td>
<td>47.6</td>
<td>48.7</td>
<td>45.6</td>
<td>30.4</td>
</tr>
<tr>
<td>Noroozi &amp; Favaro (2016)*</td>
<td>57.1</td>
<td>56.0</td>
<td>52.4</td>
<td>48.3</td>
<td>38.1</td>
</tr>
<tr>
<td>BiGAN (ours)</td>
<td>56.2</td>
<td>54.4</td>
<td>49.4</td>
<td>43.9</td>
<td>33.3</td>
</tr>
<tr>
<td>BiGAN, 112 x 112 E (ours)</td>
<td>55.3</td>
<td>53.2</td>
<td>49.3</td>
<td>44.4</td>
<td><strong>34.8</strong></td>
</tr>
</tbody>
</table>

- The first N layers of the classifier are transferred from BiGAN's encoder and frozen
- The following layers are reinitialized and trained fully supervised
BiGAN Experiments: Transferability

Figure: Classification and Fast R-CNN detection results for the PASCAL VOC 2007 test set, and FCN segmentation results on the PASCAL VOC 2012 validation set.

<table>
<thead>
<tr>
<th>trained layers</th>
<th>Classification (% mAP)</th>
<th>FRCN Detection (% mAP)</th>
<th>FCN Segmentation (% mIU)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fc8</td>
<td>fc6-8</td>
<td>all</td>
</tr>
<tr>
<td>sup. ImageNet (Krizhevsky et al., 2012)</td>
<td>77.0</td>
<td>78.8</td>
<td>78.3</td>
</tr>
<tr>
<td>k-means (Krähenbühl et al., 2016)</td>
<td>32.0</td>
<td>39.2</td>
<td>56.6</td>
</tr>
<tr>
<td>Discriminator (D)</td>
<td>30.7</td>
<td>40.5</td>
<td>56.4</td>
</tr>
<tr>
<td>Latent Regressor (LR)</td>
<td>36.9</td>
<td>47.9</td>
<td>57.1</td>
</tr>
<tr>
<td>Joint LR</td>
<td>37.1</td>
<td>47.9</td>
<td>56.5</td>
</tr>
<tr>
<td>Autoencoder ($\ell_2$)</td>
<td>24.8</td>
<td>16.0</td>
<td>53.8</td>
</tr>
<tr>
<td>BiGAN (ours)</td>
<td>37.5</td>
<td>48.7</td>
<td>58.9</td>
</tr>
<tr>
<td>BiGAN, 112 × 112 E (ours)</td>
<td>41.7</td>
<td>52.5</td>
<td>60.3</td>
</tr>
</tbody>
</table>

- Classification models are trained with various portions of the AlexNet transferred from BiGAN’s encoder and frozen.
- For detection and semantic segmentation, the BiGAN’s encoder is used as the initialization for Fast R-CNN and FCN, replacing the AlexNet model trained fully supervised for ImageNet classification.