Improving Generative Adversarial Networks with Denoising Feature Matching

David Warde-Farley, Yoshua Bengio

ICLR 2017
Improve GANs in an unsupervised way

- Observation in realistic manifestations of training procedure
  - The discriminator’s gradients w.r.t. a sample does not point in direction of a draw from the data distribution, but points in direction of local improvements (the discriminator’s estimate)
    \[
    \max_G \mathbb{E}_{z \sim p_z(z)} \left[ \log D(G(z)) \right]
    \]

- But the discriminator is a valuable source of compact descriptors
  - Versatility of high-level features learned by convolutional neural networks
Comparing with Feature Matching

- Feature matching from Improved GANs, Sailsmans et al., 2016
  train generator to match the expected value of the features
  \[
  \arg \min_{\theta_G} \left\| \mathbb{E}_{x \sim \mathcal{D}} [\Phi(x)] - \mathbb{E}_{z \sim p(z)} [\Phi(G(z))] \right\|^2
  \]
  - First moment matching in the feature space \( \Phi(\cdot) \)
  - Insensitive to higher-order statistics

- Not just match expected value but estimate and track the distribution of features (data) with denoising auto-encoder
Denoising Auto-encoder

Data distribution $p(x)$

Corrupted distribution $p(\tilde{x})$

Optimal denoising $\tilde{x} \rightarrow g(\tilde{x})$
Denoising Feature Matching

- Train DAE on the transformed training data $h = \Phi(x)$ with $x \sim \mathcal{D}$

- $r(\Phi(x')) - \Phi(x')$ with $x' = G(z)$ indicates in which direction $x'$ changed

- Minimizing $||r(\Phi(x')) - \Phi(x')||^2$ would push $x'$ towards higher probability configurations according to the data distribution in the feature space $\Phi(x)$

- Procedure
  1. Evaluate the discriminator features $\Phi(x)$
  2. Train DAE in the feature space
  3. DAE’s output on samples from the generator
  4. Treat DAE’s output as a fixed target for the generator
Denoising Feature Matching

- Denoiser objective

\[
\arg \min_{\theta_r} \mathbb{E}_{x \sim \mathcal{D}} \| \Phi(x) - r(C(\Phi(x))) \|^2
\]

- Generator objective

\[
\arg \min_{\theta_G} \mathbb{E}_{z \sim p(z)} \left[ \lambda_{\text{denoise}} \| \Phi(G(z)) - r(\Phi(G'(z))) \|^2 - \lambda_{\text{adv}} \log D(G(z)) \right]_{\text{fixed}}
\]

where denoising auto-encoder \( r(\cdot) \)
discriminator \( D = d \circ \Phi \)
feature extractor \( \Phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^k \)
classifier \( d(\cdot) : \mathbb{R}^k \to [0, 1] \)
corruption function \( C(\cdot) : \mathbb{R}^k \to \mathbb{R}^k \)
## CIFAR-10 Experiments

<table>
<thead>
<tr>
<th>Real data*</th>
<th>Semi-supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Improved GAN (Salimans et al)*</td>
<td>ALI (Dumoulin et al)†</td>
</tr>
<tr>
<td>11.24 ± .12</td>
<td>8.09 ± .07</td>
<td>5.34 ± 0.05</td>
</tr>
<tr>
<td>Ours</td>
<td>7.72 ± 0.13</td>
<td></td>
</tr>
</tbody>
</table>

Inconsistent labels due to large object size.
# STL-10 Experiments

<table>
<thead>
<tr>
<th></th>
<th>Real data</th>
<th>Ours</th>
<th>GAN Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.08</td>
<td>8.51 ± 0.13</td>
<td>7.84 ± 0.07</td>
<td></td>
</tr>
</tbody>
</table>
CIFAR-10 Experiments

<table>
<thead>
<tr>
<th></th>
<th>Real data</th>
<th>Radford et al*</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25.78 ± 0.47</td>
<td>8.83 ± 0.14</td>
<td>9.18 ± 0.13</td>
</tr>
</tbody>
</table>

*Inconsistent labels due to large object size*
Thank you
Inception Score

- Automatic method to evaluate samples
- Apply the Inception model to get $p(y|x)$
- Image contains meaningful objects
  - $H[p(y|x)]$ should be low
- The generator should generate varied images
  - $H \left[ \int p(y|x = G(z)) \, dz \right]$ should be high

$$\exp \left( \mathbb{E}_x \text{KL} \left( p(y|x) \parallel p(y) \right) \right)$$