Conditional Generative Adversarial Networks

CAMERON FABBRI
Outline

• Generative Adversarial Networks (GANs)
• Wasserstein GANs
• Conditional GANs
• Experiments
• Future Work
Generative Adversarial Networks (GANs)

- Two player mini-max game
  - Generator $G$
  - Discriminator $D$
- $D$ is trained to discriminate between a real image and a generated image.
- $G$ is trained to fool $D$

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

[1]
Generative Adversarial Networks

\[ z = \mathcal{N}(-1, 1) \]
Deep Convolutional GANs (DCGANs)

• Developed an architecture combining GANs and deep learning
Wasserstein GANs (WGAN)

• The original GAN used $D$ as a classifier, minimizing the Jensen-Shannon Divergence:

$$JS(P_r, P_g) = KL(P_r || P_m) + KL(P_g || P_m)$$

• The Wasserstein GAN applies the Kantorovich-Rubinstein duality to approximate the Wasserstein-1 distance:

$$\min_G \max_D \mathbb{E}_{x \sim P_r} \left[ D(x) \right] - \mathbb{E}_{\tilde{x} \sim P_g} \left[ D(\tilde{x}) \right]$$
Improved Wasserstein GANs (WGAN-GP)

• A differentiable function is 1-Lipschitz if and only if it has gradients with norm less than or equal to 1 everywhere
  • Idea: Enforce the Lipschitz-1 constraint by penalizing the gradient norm of the discriminator with respect to the input – not tractable.
  • Instead penalize the square distance of samples from 1.

\[ L = \mathbb{E}_{\tilde{x} \sim P_g} [D(\tilde{x})] - \mathbb{E}_{x \sim P_r} [D(x)] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]. \]

[4]
Conditional GANs (cGANs)

• Introduce some additional information into the generator and discriminator.
Experiments

• Implement the Improved Wasserstein method in the conditional GAN setting.

• Control digit generation using MNIST

• Alter visual attributes in faces using the CelebA dataset

• Explore the latent space
MNIST Experiments

\[ y = [0, 0, 0, 0, 0, 0, 1, 0, 0] \]
\[ z = \mathcal{N}(-1, 1) \]
### Label Alteration

- Each row has the same $z$ vector
- Each column has the same $y$ vector

<table>
<thead>
<tr>
<th>$z_1$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_2$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>$z_3$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>$z_4$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>$z_5$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
Interpolation

• Interpolate between both z and y vectors
CelebA Experiments

\[ z = \mathcal{N}(-1, 1) \]
\[ y = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \]
Bald, Bangs, Black Hair, Blonde Hair, Glasses, Makeup, Male, Pale, Smile
Attribute Alteration

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>Image 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bald</td>
<td>![Bald Image 1]</td>
<td>![Bald Image 2]</td>
<td>![Bald Image 3]</td>
<td>![Bald Image 4]</td>
</tr>
</tbody>
</table>
Interpolation

Bald → Not Bald

Black Hair → Blonde Hair

Female → Male

Glasses, Smiling, Bangs → No Glasses, Not Smiling, No Bangs, Pale
CelebA Experiments using WGAN-GP

- Male, Black Hair, Smiling
- Female, Smiling
- Female, Makeup
- Male, Blonde Hair
CelebA Experiments using GAN

Male, Black Hair, Smiling

Female, Smiling

Female, Makeup

Male, Blonde Hair
Conclusion

• Generative Adversarial Networks
  • Wasserstein Method
• Conditional GANs
• Experiments on two datasets
• Future Work
  • Try on other datasets
• Compare with other GAN methods
  • Least Squares GANs
  • Energy Based GANs
References


Thank you

Questions?

Code: https://github.com/cameronfabbri/cWGANs