Boundary-Seeking Generative Adversarial Networks (BGANs)

Hjelm, R. Devon, et al.

Presenting: Yevgeny Tkach

https://qdata.github.io/deep2Read/
Executive Summary

• BGAN is framework that allows GAN to generate both discrete and continuous data

• Discriminator is trained by maximizing the f-divergence between the data and generated distributions

• Generator is trained to minimize the f-divergence between the generated distribution and a self-normalized importance sampling (SIS) estimation of the data distribution

• Experiments show state of the art results in training GANs on discrete data generation and high stability in training GANs with continuous data.
Outline

• GAN – Basic Idea
• f - GAN Introduction
• Importance Sampling - Detour
• BGAN
Basic Idea of GAN

• The data we want to generate has a distribution $P(x)$
Basic Idea of GAN

• A generator G is a network. The network defines a probability distribution.

It is difficult to compute $Q(x)$
We can only sample from the distribution.

https://blog.openai.com/generative-models/
Basic Idea of GAN

1. **D** tries to output 1
2. Differentiable function **D**
3. **x** sampled from data

1. **D** tries to output 0
2. Differentiable function **D**
3. **x** sampled from model
4. Differentiable function **G**
5. Input noise **Z**
GAN Intuition

Poorly fit model

After updating D

After updating G

Mixed strategy equilibrium
GAN Formally

- **Value Function:**
  \[
  V(\mathbb{P}, G_\theta, D_\phi) = E_{x \sim P}[\log D(x)] + E_{x \sim Q}[\log(1 - D(x))] \\
  = E_{x \sim P}[\log D(x)] + E_{z \sim h(z)}[\log (1 - D(G(z)))]
  \]

- **Monte-Carlo Approximation:**
  \[
  \tilde{V}(\mathbb{P}, G_\theta, D_\phi) = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) + \frac{1}{m} \sum_{i=1}^{m} \log (1 - D(G(z^i)))
  \]

- **Discriminator target:**
  \[
  \max_{\phi} \tilde{V}(\mathbb{P}, G_\theta, D_\phi)
  \]

- **Generator target:**
  \[
  \min_{\theta} \max_{\phi} \tilde{V}(\mathbb{P}, G_\theta, D_\phi)
  \]
Algorithm

\[
\text{Initialize } \phi_d \text{ for D and } \theta_g \text{ for G}
\]

In each training iteration:

- Sample \( m \) examples \( \{x^1, x^2, \ldots, x^m\} \) from data distribution \( P(x) \)
- Sample \( m \) noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( h(z) \)
- Obtaining generated data \( \{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\} \), \( \tilde{x}^i = G(z^i) \)
- Update discriminator parameters \( \theta_d \) to maximize
  \[
  \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) + \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 - D(\tilde{x}^i) \right)
  \]
  \[\phi_d \leftarrow \phi_d + \eta \nabla \tilde{V}(\phi_d)\]
- Sample another \( m \) noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( P_{\text{prior}}(z) \)
- Update generator parameters \( \theta_g \) to minimize
  \[
  \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) + \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 - D \left( G(z^i) \right) \right)
  \]
  \[\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)\]
f - GAN Introduction


• One sentence: you can use any f–divergence
**$f$-divergence**  

$P$ and $Q$ are two distributions. $p(x)$ and $q(x)$ are the density functions respectively.

\[
D_f(P||Q) = \int q(x)f\left(\frac{p(x)}{q(x)}\right)dx
\]

- $f$ is convex
- $f(1) = 0$

Every convex function $f$ has a conjugate function $f^*$

\[
f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\} \quad \longleftrightarrow \quad f(x) = \max_{t \in \text{dom}(f^*)} \{xt - f^*(t)\}
\]
Connection with GAN

\[ f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\} \quad \leftrightarrow \quad f(x) = \max_{t \in \text{dom}(f^*)} \{xt - f^*(t)\} \]

\[ D_f(P||Q) = \int_x q(x) f \left( \frac{p(x)}{q(x)} \right) dx \]

\[ = \int_x q(x) \left( \max_{t \in \text{dom}(f^*)} \left\{ \frac{p(x)}{q(x)} t - f^*(t) \right\} \right) dx \]

D is a function whose input is x, and output is t

\[ \geq \max_{D \in \mathcal{D}} \int_x q(x) \left( \frac{p(x)}{q(x)} D(x) - f^*(D(x)) \right) dx \]

\[ = \max_{D \in \mathcal{D}} \int_x p(x) D(x) dx - \int_x q(x) f^*(D(x)) dx \]
Connection with GAN

\[ D_f(P||Q) \geq \max_D \left\{ \int p(x)D(x)dx - \int q(x)f^*(D(x))dx \right\} \]

\[ = \max_D \left\{ E_{x\sim P}[D(x)] - E_{x\sim Q}[f^*(D(x))] \right\} \]

\[ \text{Samples from } P \quad \text{Samples from } Q \]

\[ D_f(P||Q) \geq \max_D \left\{ E_{x\sim P}[\nu \circ D(x)] - E_{x\sim Q}[f^*(\nu \circ D(x))] \right\} \]

\[ G^* = \arg \min_G D_f(P||Q) \]

\[ = \arg \min_G \max_D \left\{ E_{x\sim P}[\nu \circ D(x)] - E_{z\sim h(z)}[f^*(\nu \circ D(G(z)))] \right\} \]

GAN value function:

\[ V(\mathbb{P}, G_\theta, D_\phi) = E_{x\sim P}[\log D(x)] + E_{z\sim h(z)}[\log (1 - D(G(z)))] \]
Importance Sampling - Detour

\[ E_{x \sim p}[f(x)] = \int f(x)p(x)dx \]

\[ = \int f(x)\frac{p(x)}{q(x)}q(x)dx \]

\[ = \int f(x)w(x)q(x)dx \]

\[ = \frac{E_{x \sim q}[f(x)w(x)]}{E_{x \sim q}[w(x)]} \]

\[ w(x) = \frac{p(x)}{q(x)} \]

In case \( p \) or \( q \) are scaled density functions

\( w(x) \) - Importance Weights
Boundary Seeking GAN - BGAN

Theorem 1: \( P \) and \( Q \) as in f-GAN, and \( D^* \in D \) satisfying:

\[
D_f(P \| Q) = \max_D \{E_{x \sim P}[D(x)] - E_{x \sim Q}[f^*(D(x))]\}
\]

Then: \( p(x) = \left(\frac{\partial f^*}{\partial D}\right)(D^*(x))q(x) \)

Proof:

\[
D_f(P \| Q) = E_{x \sim Q} \left[ f \left( \frac{p(x)}{q(x)} \right) \right] = E_{x \sim Q} \left[ \sup_t \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} \right]
\]

\( p \) re-written in terms of \( q \) and a scaling factor
\[
w(x) = \left(\frac{\partial f^*}{\partial D}\right)(D^*(x)) \quad \text{ – Importance weights}
\]
Boundary Seeking GAN - BGAN

BGAN suggests to use the divergence between \( q(x) \) and the self normalized importance sampling (IS) estimation of \( p(x) \):

\[
\tilde{p}(x) = \frac{w(x)}{\beta} q(x)
\]

Where:

\[
\beta = E_{x \sim Q}[w(x)]
\]
BGAN – IS intuition

- Divergence between ▲ should have lower variance than if taking arbitrary samples from $P(x)$
- Since $G(z)$ defines a distribution that $x$ is sampled from - the variance can be further decreased by taking multiple samples from the same $z$
BGAN – reduced variance

We can restate everything in terms of conditional distributions:

- $q(x) = \int_Z g(x|z)h(z)dz$
- $g(x|z): Z \rightarrow [0,1]^d$ - multivariate Bernoulli distribution
- $\alpha(z) = E_{x \sim g(x|z)}[w(x)]$ - similar to $\beta$
- $\tilde{p}(x|z) = \frac{w(x)}{\alpha(z)} g(x|z)$
- $D_{KL}(\tilde{p}(x)||q(x)) = E_{h(z)}[D_{KL}(\tilde{p}(x|z)||q(x|z))]$
- $\nabla E_{h(z)}[D_{KL}(\tilde{p}(x|z)||q(x|z))]$ approximates with two MC
BGAN - Algorithm

Algorithm 1. Discrete Boundary Seeking GANs

\[(\theta, \phi) \leftarrow \text{initialize the parameters of the generator and statistic network}\]

repeat

\[\hat{x}^{(n)} \sim \mathbb{P}\]

\[z^{(n)} \sim h(z)\]

\[x^{(m|n)} \sim g_\theta(x | z^{(n)})\]

\[\triangleright \text{Draw } N \text{ samples from the empirical distribution}\]

\[\triangleright \text{Draw } N \text{ samples from the prior distribution}\]

\[\triangleright \text{Draw } M \text{ samples from each conditional } g_\theta(x | z^{(m)}) \text{ (drawn independently if } \mathbb{P} \text{ and } \mathbb{Q}_\theta \text{ are multi-variate)}\]

\[w(x^{(m|n)}) \leftarrow (\partial f^* / \partial T) \circ (\nu \circ F_\phi(x^{(m|n)}))\]

\[\tilde{w}(x^{(m|n)}) \leftarrow w(x^{(m|n)}) / \sum_{m'} w(x^{(m'|n)})\]

\[\triangleright \text{Compute the un-normalized and normalized importance weights (applied uniformly if } \mathbb{P} \text{ and } \mathbb{Q}_\theta \text{ are multi-variate)}\]

\[\mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, T_\phi) \leftarrow \frac{1}{N} \sum_n F_\phi(\hat{x}^{(n)}) - \frac{1}{N} \sum_n \frac{1}{M} \sum_m w(x^{(m|n)})\]

\[\triangleright \text{Estimate the variational lower-bound}\]

\[\phi \leftarrow \phi + \gamma_d \nabla_\phi \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, T_\phi)\]

\[\triangleright \text{Optimize the discriminator parameters}\]

\[\theta \leftarrow \theta + \gamma_g \frac{1}{N} \sum_{n,m} \tilde{w}(x^{(m|n)}) \nabla_\theta \log g_\theta(x^{(m|n)} | z)\]

\[\triangleright \text{Optimize the generator parameters}\]

until convergence
Boundary Seeking GAN - BGAN

\[ D_f(P_{data} || P_G) \geq \max_D \{ E_{x \sim P_{data}} [\nu \circ D(x)] - E_{x \sim P_G} [f^*(\nu \circ D(x))] \} \]

\[ \hat{p}(x) = \frac{w(x)}{\beta} q(x) \quad w(x) = (\frac{\partial f^*}{\partial D})(D^*(x)) \]

Table 1: Important weights and nonlinearities that ensure

<table>
<thead>
<tr>
<th>Importance weights for ( f )-divergences</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )-divergence</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>GAN</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
</tr>
<tr>
<td>KL</td>
</tr>
<tr>
<td>Reverse KL</td>
</tr>
<tr>
<td>Squared-Hellinger</td>
</tr>
</tbody>
</table>
BGAN – Experiments

<table>
<thead>
<tr>
<th>Train Measure</th>
<th>Eval Measure (lower is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JS</td>
</tr>
<tr>
<td>BGAN - JS</td>
<td>0.37 (±0.02)</td>
</tr>
<tr>
<td>BGAN - reverse KL</td>
<td>0.44 (±0.02)</td>
</tr>
<tr>
<td>WGAN-GP (samples)</td>
<td>0.45 (±0.03)</td>
</tr>
<tr>
<td>WGAN-GP (softmax)</td>
<td>-</td>
</tr>
</tbody>
</table>
BGAN – Experiments

And it ’s miant a quert could he
” We paite of condels of money wi
Lankard Avaloma was Mr. Palin ,
Thene says the sounded Sunday in
About dose and warthestrinds fro

He weirist placed produces hopesi
Sance Jory Chorotic , Sen doesi
What was like one of the July 2
The BBC nothing overton and slea
College is out in contesting rev
BGAN – Continuous case

Recall:

\[ G^* = \arg \min_G D_f(P_{data} \| P_G) \]

\[ D_f(P \| Q) = \mathbb{E}_{x \sim Q} \left[ f \left( \frac{p(x)}{q(x)} \right) \right] = \mathbb{E}_{x \sim Q} \left[ \sup_t \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} \right] \]

\[ \updownarrow \quad \text{Max when } \nabla \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} = 0 \]

\[ \frac{p(x)}{q(x)} = \left( \frac{\partial f^*}{\partial D} \right)(D^*(x)) = w(x) \]

\[ \updownarrow \quad p(x) = q(x) \text{ when } w(x) = 1 \]

\[ G^* = \arg \min_G \left( \log w(G(z)) \right)^2 \]

\[ \updownarrow \]

\[ G^* = \arg \min_G D(G(z))^2 \]
BGAN – Continuous case

\(f\)–GAN:
\[
G^* = \arg \min_G \{E_{x \sim P} [\nu \circ D(x)] - E_{z \sim h(z)} [f^* (\nu \circ D(G(z)))]\}
\]

GAN (Proxy GAN):
\[
G^* = \arg \min_G \left\{ E_{x \sim P} [\log D(x)] + E_{z \sim h(z)} \left[ \log \left(1 - D(G(z)) \right) \right]\right\}
\]

BGAN:
\[
G^* = \arg \min_G E_{z \sim h(z)} D(G(z))^2 \quad \iff \quad w(x) = 1 \quad \iff \quad p(x) = q(x)
\]
BGAN – Continuous Experiments

Figure 3: Highly realistic samples from a generator trained with BGAN on the CelebA and LSUN datasets. These models were trained using a deep ResNet architecture with gradient norm regularization (Roth et al., 2017). The Imagenet model was trained on the full 1000 label dataset without conditioning.
BGAN – Continuous Experiments

- Generator trained for 5 steps for every 1 step of the discriminator
BGAN – Continuous Experiments

- Train a DCGAN using the proxy loss.
- Train the discriminator for 1000 more steps
- Perform gradient descent directly on the pixels

Starting image (generated)
BGAN – Continuous Experiments
Discussion