Energy-Based Generative Adversarial Network

Energy-Based Generative Adversarial Network
J. Zhao, M. Mathieu and Y. LeCun

Learning to Draw Samples: With Application to Amoritized MLE for Generalized Adversarial Learning
D. Wang and Q. Liu

Presented by Chunyuan Li
Preliminaries: “vanilla” GAN

1. Real data distribution \( P_r(x) \), where \( x \in \mathcal{X} \);

2. Fake data distribution \( P_g(x) \), implemented via a generator as \( x = G(z), z \sim P(z) \), where \( z \in \mathcal{Z} \);

- **Objective**

\[
\min_G \max_D V(D, G)
\]

- **Discriminator** \( D : \mathcal{X} \rightarrow [0, 1] \)

\[
\mathcal{L}_d = -\mathbb{E}_{x \sim P_r}[\log D(x)] - \mathbb{E}_{x \sim P_g}[\log(1 - D(x))] \tag{1}
\]

\( D(x) \): the **probability** that \( x \) from the real data rather than generator.

- **Generator** \( G : \mathcal{Z} \rightarrow \mathcal{X} \)

\[
\mathcal{L}_g = \mathbb{E}_{x \sim P_g}[\log(1 - D(x))] = \mathbb{E}_{z \sim P(z)}[\log(1 - D(G(z)))] \tag{2}
\]
Ideas of the two papers

1. **EBGAN**
   - An energy-based formulation for adversarial training
   - Autoencoder is employed for the energy function
   - Repelling force

2. **SteinGAN**
   - Amortized SVGD algorithm
   - A probabilistic version of EBGAN
EBGAN

- **Discriminator** $D : \mathcal{X} \rightarrow \mathbb{R}$.

\[
\mathcal{L}_d = \mathbb{E}_{x \sim P_r}[D(x)] + \mathbb{E}_{x \sim P_g}[\max(0, m - D(x))] \quad (3)
\]

where $m$ is a positive margin.

- **Generator** $G : \mathcal{Z} \rightarrow \mathcal{X}$

\[
\mathcal{L}_g = \mathbb{E}_{x \sim P_g}[D(x)] = \mathbb{E}_{z \sim P(z)}[D(G(z))] \quad (4)
\]
EBGAN: Autoencoder as the energy function

\[ D(x) = \| \text{Dec}(\text{Enc}(x)) - x \| \] \hspace{1cm} (5)

- **Discriminator**

\[ \mathcal{L}_d = \mathbb{E}_{x \sim P_r}[\text{Dec}(\text{Enc}(x)) - x] \]
\[ + \mathbb{E}_{x \sim P_g}[\max(0, m - \| \text{Dec}(\text{Enc}(x)) - x \|)] \] \hspace{1cm} (6)

- **Generator**

\[ \mathcal{L}_g = \mathbb{E}_{z \sim P(z)}[\| \text{Dec}(\text{Enc}(G(z))) - G(z) \|] \] \hspace{1cm} (7)

**Figure**: EBGAN architecture.
Repelling regularizer

\[ \mathcal{L}_{PT}(S) = \frac{1}{bs(bs - 1)} \sum_i \sum_{j \neq i} \left( \frac{S_i^\top S_j}{\|S_i\| \|S_j\|} \right) \]  

- \( bs \): batch size
- \( S_i \): \( i \)-th sample in minibatch \( S \) on feature level \( \text{Enc}(G(z)) \)

Decreasing similarity between the sample representations within the minibatch and thus makes them as mutually orthogonal as possible, and further increases the diversity of generated samples

Regularized generator loss

\[ \mathcal{L}'_g = \mathcal{L}_g + \lambda \mathcal{L}_{PT}, \quad \lambda = 0.1 \]
SteinGAN: SVGD for $P$

- Idea: approximate target distribution with a set of particles
- SVGD updates the particles iteratively $x_i \leftarrow x_i + \epsilon \phi(x_i)$, where $\phi(x)$ is a “particle gradient direction” chosen to maximumly decrease the KL divergence:

$$\phi = \arg\max_{\phi \in F} \left\{ -\frac{d}{d\epsilon} KL(P_g^{[\epsilon\phi]} || P)|_{\epsilon=0} \right\}$$

where

- $P_g^{[\epsilon\phi]}$ denotes the approximated density of the updated particle
- $F$ is the set of perturbation directions that we optimize over
- When $F$ is RKHS, there exists a closed solution, approximated with $n$ particles

$$\phi^*(x_i) \approx \Delta x_i = \frac{1}{n} \sum_{i=1}^{n} [\nabla_x \log p(x)k(x, x_i) + \nabla_x k(x, x_i)]$$

- 1st term: drive toward the high probability regions of $P_r$
- 2nd term: repulsive force to encourage diversity
Amortize SVGD

- Disadvantage of SVGD: Generation/inference can not improve based on the experience from the past tasks.
- Solution: Training a $\eta$-parameterized neural network $f(\eta; z), z \sim \mathcal{N}(0, \mathbf{I})$ to mimic the SVGD dynamics.
- An incremental approach in which $\eta$ is iteratively adjusted so that the change of network outputs fit the change in SVGD.

$$
\eta^{t+1} \leftarrow \arg\min_{\eta^{t+1}} \sum_{i=1}^{m} \| f(\eta^{t+1}; z_i) - f(\eta^{t}; z_i) - \epsilon \Delta x_i \|^2 
$$  \hspace{1cm} (12)

- Combined with Taylor expansion

$$
f(\eta^{t+1}; z_i) - f(\eta^{t}; z_i) \approx \nabla_\eta f(\eta^{t}; z_i)(\eta^{t+1} - \eta^{t})
$$

$$
\eta^{t+1} \leftarrow \eta^{t} + \epsilon \sum_{i=1}^{m} \nabla_\eta f(\eta^{t}; z_i) \Delta x_i
$$  \hspace{1cm} (13)

This is a “chain rule” that backpropagates the Stein variational gradient to the network parameter $\eta$. 

Maximum likelihood with energy functions

- The empirical data distribution as \( P_r \), and we build a \( \omega \)-parameterized \( p(x|\omega) \) as our model, with the energy function:

\[
p(x|\omega) = \exp\{-f(x, \omega) - F(\omega)\} \quad \text{and} \quad F(\omega) = \log \int_{p_\omega(x)} \exp\{-f(x, \omega)\} \quad (14)
\]

where \( p_\omega(x) \) is the marginal \( x \) under our model.

- The MLE for the entire dataset is:

\[
\max_{\omega} \{ L(\omega; x) = \int_{P_r} \log p(x|\omega) \} \quad (15)
\]

Gradient: \( \nabla_\omega L = -\int_{P_r} [\nabla_\theta f(x, \omega)] + \int_{p_\omega(x)} [\nabla_\omega f(x, \omega)] \quad (16) \)

- The expectation of 2nd term is intractable, \( P_g \) is used as its approximation.
Amortize SVGD for GAN

Reformulate SteinGAN with the notations of EBGAN

\[ D(x) = f(\omega, x) \quad \text{and} \quad G(z) = f(\eta, z) \quad (17) \]

1. Discriminator (Update \( \omega \) with SGD):

\[ \mathcal{L}_d = \mathbb{E}_{x \sim P_r}[D(x)] - \mathbb{E}_{x \sim P_g}[D(x)] \quad (18) \]

Decreases the energy of the true data and increases the energy of the simulated data

2. Generator (Update \( \eta \) with Amortized SVGD):

\[ \mathcal{L}_g = \mathbb{E}_{x \sim P_g}[D(x)] = \mathbb{E}_{z \sim P(z)}[D(G(z))] \quad (19) \]

Decreases the energy of the simulated data to better fit with \( p(x|\omega) \).

3. Specification of \( D(x) \) and \( k(x, x') \)

\[ D(x) = \|\text{Dec(Enc}(x)) - x\| \quad (20) \]

\[ k(x, x') = \exp\left(-\frac{1}{h^2}\|\text{Enc}(x) - \text{Enc}(x')\|^2\right), \text{~} h \text{ is bandwidth} \quad (21) \]
Inception Score (IS): Evaluation Metric

- Pretrained Inception model $p(y|x)$ is applied to every generated image.

- C1: For higher quality of a single image, the conditional $p(y|x)$ with lower entropy is expected.

- C2: For higher diversity of all images, the marginal $\hat{p}(y) = \mathbb{E}_{z \sim P(z)} p(y|x = G(z))$ with higher entropy is expected.

- Combined C1 and C2, IS is obtained as

$$\exp(\mathbb{E}_{P_g}[KL(p(y|x) || \hat{p}(y))])$$

(22)
## Preliminaries

### EBGAN
- MNIST dataset
- Test various architectures and optimization methods (512 experiments)

### SteinGAN
- Cifar10 dataset

<table>
<thead>
<tr>
<th>Real Training Set</th>
<th>DCGAN</th>
<th>SteinGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.848</td>
<td>7.368</td>
<td>7.428</td>
</tr>
</tbody>
</table>

![Graph showing inception scores for EBGAN and SteinGAN]