An Introduction to Generative Adversarial Nets and Application to NRI image data

Xin Zhang  Haozhe Zhang  Zhengyuan Zhu

Iowa State University

10/14/2017
1 Background

2 Generative Adversarial Nets

3 Conditional Adversarial Networks

4 Preliminary Results for NRI images

5 References
This project is a part of the National Resources Inventory (NRI);

Main goal:
- Detecting new roads from the satellite image;
- Classifying land usage and detecting usage changes;
- ...
1 Background

2 Generative Adversarial Nets

3 Conditional Adversarial Networks

4 Preliminary Results for NRI images

5 References
Generative Adversarial Nets is inspired by a minimax two-player game.

**Discriminative model:**
- Binary classifier;
- Estimate the probability that a given sample from the training data distribution;
- Determine whether a given sample if from the data distribution or the model distribution;

**Generative model:**
- Generator;
- Capturing the distribution of the training data
- A differentiable function to map the input noise into a "fake training data";

Usually, Generative model (G) and Discriminative model (D) are two non-linear function, such as multilayer preceptron ...
if the data is from the training data (true data), then the output of D will be close to 1;

if the data is a noise, then the generator G will generate a fake data, and the discriminator D will output a value close to 0.

Notation: the distribution of training data as $p_{data}$; the distribution of prior noise as $p_z$; the distribution of generated samples on prior noise as $p_G$;

Define generative model as $G(.|\theta_G)$ and discriminative model as $D(.|\theta_D)$;

Aim: fixed $G$, train $D$ to maximize the probability that it could correctly tell whether a sample is from $p_{data}$; given $D$, train $G$ so that for any generated sample $x'$, $D(x')$ is close to 1;
Model (Con’t)

In GANs, there is no loss function. Instead, a value function is proposed to train $D$ and $G$:

$$(G^*, D^*) = \arg \min_G \max_D V(D, G)$$

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

Where $G^*$ is the generative model we want and $D^* = \frac{1}{2}$
Algorithm (Goodfellow et al. (2014))

Algorithm 1 Minibatch SGD training of generative adversarial nets

1: for number of training iterations do
2:     for k steps do
3:         Sample minibatch of $m$ noise samples $\{z^{(i)}\}_{i=1}^m$ from $p_z$;
4:         Sample minibatch of $m$ examples $\{x^{(i)}\}_{i=1}^m$ from $p_{data}$;
5:         Update the discriminator by ascending its stochastic gradient:
6:             \[ \nabla \theta_D \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]; \]
7:     EndFor
8:     Sample minibatch of $m$ noise samples $\{z^{(i)}\}_{i=1}^m$ from $p_z$;
9:     Update the generator by descending its stochastic gradient:
10: \[ \nabla \theta_G \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^{(i)}))); \]
9:     EndFor
Theoretical Results

Proposition
For $G$ fixed, the optimal discriminator $D$ is

$$D^*_G(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}.$$ 

Theorem
The global minimum of the virtual training criterion $C(G) = \max_D V(G, D)$ is achieved if and only if $p_G = p_{\text{data}}$. At that point, $C(G)$ achieves the value $-\log 4$. 
Theoretical Results (Con’t)

Proposition

If $G$ and $D$ have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given $G$, and $p_G$ is updated so as to improve the criterion

$$
\mathbb{E}_{x \sim p_{\text{data}}} [\log D_G^*(x)] + \mathbb{E}_{x \sim p_G} [\log (1 - D_G^*(x))],
$$

where $D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$, then $p_G$ converges to $p_{\text{data}}$. 

Zhang & Zhang & Zhu
GANs
1 Background

2 Generative Adversarial Nets

3 Conditional Adversarial Networks

4 Preliminary Results for NRI images

5 References
Idea

- **Motivation:** the unconditional GANs is too "free":
  - no assumption for data distribution $p_{data}$;
  - with highly-resolution images, the generative model could not be controlled;
- **Solution:** giving some auxiliary information to direct the data generation process;
- **GANs** (unsupervised learning) $\rightarrow$ **CGANs** (supervised learning)
Model

The value function for CGANs (Mirza and Osindero (2014)):

\[(G^*, D^*) = \arg \min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x|y)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z|y)))]\]

Figure: the structure for CGANs, cited from Mirza and Osindero (2014)
Remaining the discriminator’s task, tasking generator to generate output near the ground truth in $L_p$ sense, as well as fool the discriminator. New value functions are given by adding $L_2$ penalty (Pathak et al. (2016)) or $L_1$ penalty (Isola et al. (2016))

$$(G^*, D^*) = \arg \min_G \max_D V_{CGANs}(G, D) + \lambda \mathbb{E}_{x \sim p_{data}, z \sim p_z} \|x - G(z | y)\|_p$$
Experiment Result

Figure: Experiment Results from Isola et al. (2016)
1. Background

2. Generative Adversarial Nets

3. Conditional Adversarial Networks

4. Preliminary Results for NRI images

5. References
Training Images (Isola et al. (2016))
Results: Urban

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Urban

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Urban

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Suburb

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Suburb

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
**Results: Suburb**

**Figure:** (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Suburb

Figure: (1,1) is the original; (1,2) is with total-trained; (2,1) is with city-trained; (2,2) is with suburb-trained.
Results: Urban Crops

Figure: The first column is the original data; the second is from city-trained; the third is from suburb-trained.
Results: Suburb Cops

**Figure:** The first column is the original data; the second is from city-trained; the third is from suburb-trained.
Discussion

- **Model**
  - redesign the structure of the network;
  - properly add some prior information;
  - ...

- **Training data**
  - need to produce classified training images (by states);
  - highly resolution training images;
  - ...

- **Testing data**
  - providing the shooting time (additional information);
  - resized issue;
  - ...
1 Background

2 Generative Adversarial Nets

3 Conditional Adversarial Networks

4 Preliminary Results for NRI images

5 References
References


