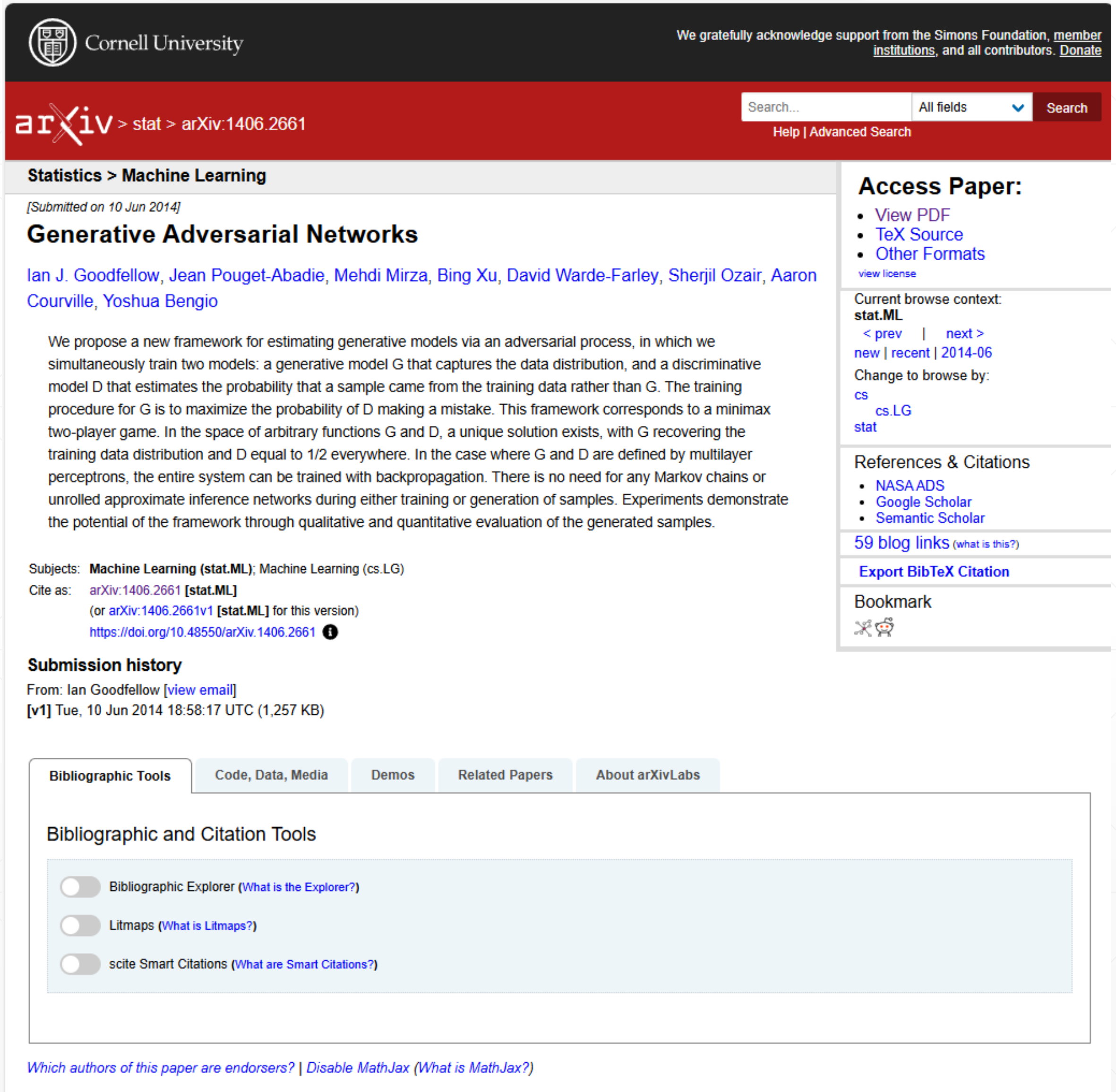
0705 Paper Review

21 박연수

Index

- 1. GAN
  - Adversarial nets
  - Theoretical Results
  - Experiments
  - Advantages and disadvantages
  - Conclusions and future work



Cornell University

We gratefully acknowledge support from the Simons Foundation, member institutions, and all contributors. Donate

arXiv > stat > arXiv:1406.2661

Search... | All fields | Search

Help | Advanced Search

Statistics > Machine Learning

[Submitted on 10 Jun 2014]

Generative Adversarial Networks

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Subjects: **Machine Learning (stat.ML)**; Machine Learning (cs.LG)
Cite as: arXiv:1406.2661 **[stat.ML]**
(or arXiv:1406.2661v1 **[stat.ML]** for this version)
https://doi.org/10.48550/arXiv.1406.2661

Submission history

From: Ian Goodfellow [view email]
[v1] Tue, 10 Jun 2014 18:58:17 UTC (1,257 KB)

Access Paper:

- View PDF
- TeX Source
- Other Formats

view license

Current browse context:
stat.ML
< prev | next >
new | recent | 2014-06
Change to browse by:
cs
cs.LG
stat

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

59 blog links (what is this?)

Export BibTeX Citation

Bookmark

Bibliographic Tools | Code, Data, Media | Demos | Related Papers | About arXivLabs

Bibliographic and Citation Tools

Bibliographic Explorer (What is the Explorer?)
Litmaps (What is Litmaps?)
scite Smart Citations (What are Smart Citations?)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In NIPS, 2014

1406.2661 (arxiv.org)

Adversarial nets

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

- G is generator mapping noise z to data space.
- D is discriminator represents the probability that x came from the data rather than pg
- G to minimize log(1 – D(G(z)))
- D to maximize log(D(x))
- D and G play the following two-player minimax game with value function V (G, D)

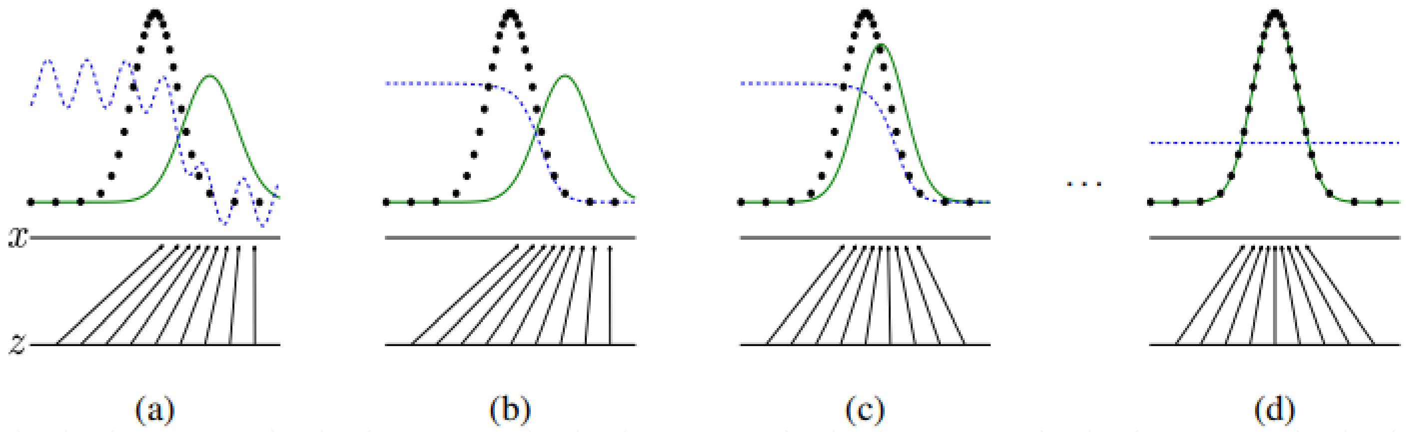
Adversarial nets

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

- Optimizing D to completion in the inner loop of training is computationally prohibitive
- k steps of optimizing D and one step of optimizing G. -> D being maintained near its optimal solution, so long as G changes slowly enough.

- G to minimize log(1 – D(G(z))) VS G to maximize log D(G(z))
  - In practice, equation may not provide sufficient gradient for G to learn well.
  - when G is poor, D can reject samples with high confidence In this case, log(1 – D(G(z))) saturates.
  - We can train G to maximize log D(G(z))

Adversarial nets



- D : blue, dashed line
- G/pg : green, solid line
- Data : black, dotted line
- Arrows : x = G(z)

- (a) Consider an adversarial pair near convergence: pg is similar to pdata and D is a partially accurate classifier.
- (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

- (c) After an update to G, gradient of D has guided G(z) to flow to regions that are more likely to be classified as data.
- (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because pg = pdata. The discriminator is unable to differentiate between the two distributions, i.e. D(x) = 1/2

Theoretical Results

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Theoretical Results

- Why KL divergence appears in this paper?

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$JSD(p\,||\,q) = \frac{1}{2}(KL(p\,||\,M) + JSD(q\,||\,M))$$
$$where,\ M = \frac{1}{2}(p+q)$$

$$\begin{aligned} C(G) =& \max_D V(G,D) \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}}\,\|p_g\right)$$

$$KL(B\,||\,A) = \mathbb{E}_{x\sim B}[log\frac{B(x)}{A(x)}]$$

let A as p_g and let B as p_{data}

$$C(G) = -log4 + \mathbb{E}_{x\sim p_{data}}[log\frac{2p_{data}(x)}{p_{data}(x)+p_g}] + \mathbb{E}_{x\sim p_z}[log\frac{2p_g(x)}{p_{data}(x)+p_g}]$$

$$= -log4 + KL(p_{data}\,||\,\frac{p_{data}+p_g}{2}) + KL(p_g\,||\,\frac{p_{data}+p_g}{2})$$

Theoretical Results

- Why KL divergence appears in this paper?

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$\begin{aligned} C(G) =& \max_D V(G, D) \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

$$KL(B\,||\,A) = \mathbb{E}_{x\sim B}[log\frac{B(x)}{A(x)}]$$

let A as p_g and let B as p_{data}

$$C(G) = -log4 + \mathbb{E}_{x\sim p_{data}}[log\frac{2p_{data}(x)}{p_{data}(x) + p_g}] + \mathbb{E}_{x\sim p_z}[log\frac{2p_g(x)}{p_{data}(x) + p_g}]$$

$$= -log4 + KL(p_{data}\,||\,\frac{p_{data} + p_g}{2}) + KL(p_g\,||\,\frac{p_{data} + p_g}{2})$$

$$JSD(p\,||\,q) = \frac{1}{2}(KL(p\,||\,M) + JSD(q\,||\,M))$$

$$where,\, M = \frac{1}{2}(p+q)$$

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}}\,\|p_g\right)$$

Theoretical Results

- Why KL divergence appears in this paper?

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$\begin{aligned} C(G) =& \max_D V(G, D) \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

$$KL(B\,||\,A) = \mathbb{E}_{x\sim B}[log\frac{B(x)}{A(x)}]$$

let A as p_g and let B as p_{data}

$$C(G) = -log4 + \mathbb{E}_{x\sim p_{data}}[log\frac{2p_{data}(x)}{p_{data}(x) + p_g}] + \mathbb{E}_{x\sim p_z}[log\frac{2p_g(x)}{p_{data}(x) + p_g}]$$

$$= -log4 + KL(p_{data}\,||\,\frac{p_{data} + p_g}{2}) + KL(p_g\,||\,\frac{p_{data} + p_g}{2})$$

$$JSD(p\,||\,q) = \frac{1}{2}(KL(p\,||\,M) + JSD(q\,||\,M))$$

$$where,\, M = \frac{1}{2}(p + q)$$

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}}\,\|p_g\right)$$

Advantages and disadvantages

Advantages	Disadvantages
1. Markov chains are never needed 2. no inference is needed 3. wide variety of functions can be incorporated into the model	1. there is no explicit representation of pg(x) 2. D must be synchronized well with G during training

Conclusions and future work

- A conditional generative model
- Learned approximate inference
- Semi-supervised learning
- Etc ...

- This paper has demonstrated the viability of the adversarial modeling framework, suggesting that these research directions could prove useful.

Thank you