4 Years of Generative Adversarial Networks (GANs)

Lu Lu

Crunch Seminar

Chinese New Year 2018
Best Wishes for the Year of the Dog
Overview

1. What is “adversarial”? 

2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
Do Not Be Afraid of GANs.

Neural network AI is simple.

By Brandon Wirtz (CEO and Founder at Recognant), Feb 15, 2018

- ... 99% of these things are completely stupid...
- **So you built a neural network from scratch And it runs on a phone**
  Great. So you converted 11 lines of python that would fit on a t-shirt... You have mastered what a cross compiler can do in 3 seconds.

What about GANs? 3%.
Overview

1. What is “adversarial”?

2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
What Is Adversarial?

Evolve with competition

Deer-leopard minimax game

\[ \min_{\text{leopard}} \max_{\text{deer}} V(\text{deer}, \text{leopard}) = \text{distance between deer and leopard} \]

What Doesn’t Kill You Makes You Stronger!
What Is Adversarial?

- Generative adversarial networks (GANs)
Overview

1. What is “adversarial”?
2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN
3. Paper review: Daskalakis, Training GANs with optimism, 2017
4. Boundary equilibrium GAN (BEGAN)
5. Adversarial examples

Since 2014,
Vanilla GAN

- **Generator G**: capture the data distribution (make realistic images)
- **Discriminator D**: estimate the probability that a sample came from the training data rather than G (tell real and fake images apart)

![Diagram of Vanilla GAN]

The generator is trained to map a noise sample to a synthetic data sample that can “fool” the discriminator.

The discriminator is trained to distinguish real data samples from synthesized samples.
Vanilla GAN

- $p_z(z)$: input noise
- $p_{\text{data}}(x)$: real data’s distribution
- $p_g(x)$: generator’s distribution of $G(z)$

Two-player minimax game

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

Main loop of GAN training
Vanilla GAN

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 $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
    • Sample minibatch of $m$ examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
    • Update the discriminator by ascending its stochastic gradient:
      \[
      \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left(1 - D \left(G \left(z^{(i)}\right)\right)\right) \right].
      \]
  end for
  • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Update the generator by descending its stochastic gradient:
    \[
    \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D \left(G \left(z^{(i)}\right)\right)\right).
    \]
end for

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

Difficult to train [Salimans, 2016, Arjovsky, 2017a]

- Vanishing gradients
- Instability
- Model collapse
- ...

Lu Lu (Crunch)  
4 years of GANs  
Chinese New Year 2018
Improvements

17 tips to make GANs work (https://github.com/soumith/ganhacks)

- Use a spherical $z$
- Batch normalization [Ioffe, 2015]
- ...

GAN variants (> 100)

- Deep convolutional GAN (DCGAN) [Radford, 2015]
- Conditional GAN [Mirza, 2014]
- Adversarially learned inference (ALI) [Dumoulin, 2016]
- Adversarial autoencoder (AAE) [Makhzani, 2015]
- Energy-based GAN (EBGAN) [Zhao, 2016]
- Wasserstein GAN (WGAN) [Arjovsky, 2017b]
- Boundary equilibrium GAN (BEGAN) [Berthelot, 2017]
- Bayesian GAN [Saatchi, 2017]
- ...
Overview

1. What is “adversarial”?  

2. Generative adversarial networks (GANs)  
   - Vanilla GAN  
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
Recall GAN

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

Earth-Mover (EM) distance or Wasserstein-1 [Monge, 1781]

$$
W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}((x, y) \sim \gamma)[||x - y||]
$$

By Kantorovich-Rubinstein duality [Villani, 2008],

$$
W(P_r, P_\theta) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_\theta}[f(x)]
$$

Discriminator $f_w$, generator $g_\theta$

$$
\min_{\theta} \max_w \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]
$$
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: $\alpha$, the learning rate. $c$, the clipping parameter. $m$, the batch size. $n_{\text{critic}}$, the number of iterations of the critic per generator iteration.

Require: $w_0$, initial critic parameters. $\theta_0$, initial generator’s parameters.

1: while $\theta$ has not converged do
2:     for $t = 0, \ldots, n_{\text{critic}}$ do
3:         Sample $\{x^{(i)}\}_{i=1}^{m} \sim P_r$ a batch from the real data.
4:         Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
5:         $g_w \leftarrow \nabla_w \left[ \frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_{\theta}(z^{(i)})) \right]$  
6:         $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
7:         $w \leftarrow \text{clip}(w, -c, c)$
8:     end for
9:     Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
10:    $g_{\theta} \leftarrow -\nabla_{\theta} \frac{1}{m} \sum_{i=1}^{m} f_w(g_{\theta}(z^{(i)}))$
11:    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_{\theta})$
12: end while
WGAN

- Improved stability of learning
- Get rid of mode collapse
- Meaningful learning curves

GAN without tricks during training

WGAN samples
Overview

1. What is “adversarial”?

2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
Problem of WGAN

Limit cycling behavior in training (W)GAN

[https://en.wikipedia.org/wiki/Limit_cycle]
Gradient Descent (GD) vs. Optimistic Mirror Descent (OMD)

**GD**

\[ w_{t+1} = w_t + \eta \cdot \nabla w, t \]
\[ \theta_{t+1} = \theta_t - \eta \cdot \nabla \theta, t \]

**OMD** [Rakhlin, 2013]: \( M_{.t+1} \) is a predictor of \( \nabla ., t \)

\[ w_{t+1} = w_t + \eta \cdot (\nabla w, t + M_{w,t+1} - M_{w,t}) \]
\[ \theta_{t+1} = \theta_t - \eta \cdot (\nabla \theta, t + M_{\theta,t+1} - M_{\theta,t}) \]

In this paper, choose \( M_{.t+1} = \nabla ., t \)

\[ w_{t+1} = w_t + \eta \cdot (2\nabla w, t - \nabla w, t-1) = w_t + 2\eta \cdot \nabla w, t - \eta \cdot \nabla w, t-1 \]
\[ \theta_{t+1} = \theta_t - \eta \cdot (2\nabla \theta, t - \nabla \theta, t-1) = \theta_t - 2\eta \cdot \nabla \theta, t + \eta \cdot \nabla \theta, t-1 \]
Gradient Descent (GD) vs. Optimistic Mirror Descent (OMD)

(a) GD dynamics.

(b) OMD dynamics.

OMD dynamics converge in terms of the last iterate.
Optimistic ADAM

ADAM (adaptive moment estimation) [Kingma, 2014] (6475 citations)

Algorithm 1: Adam, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. $g_t^2$ indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With $\beta_1^t$ and $\beta_2^t$ we denote $\beta_1$ and $\beta_2$ to the power $t$.

Require: $\alpha$: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters $\theta$
Require: $\theta_0$: Initial parameter vector

$m_0 \leftarrow 0$ (Initialize $1^{\text{st}}$ moment vector)
$v_0 \leftarrow 0$ (Initialize $2^{\text{nd}}$ moment vector)
$t \leftarrow 0$ (Initialize timestep)

while $\theta_t$ not converged do
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep $t$)
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
  $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
  $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
  $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while

return $\theta_t$ (Resulting parameters)
Optimistic ADAM

ADAM:

\[ \theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \]

where \( \hat{m}_t \) is first moment, \( \hat{v}_t \) is second moment

**Algorithm 1** *Optimistic ADAM*, proposed algorithm for training WGANs on images.

Parameters: stepsizes \( \eta \), exponential decay rates for moment estimates \( \beta_1, \beta_2 \in [0, 1] \), stochastic loss as a function of weights \( \ell_t(\theta) \), initial parameters \( \theta_0 \)

**for** each iteration \( t \in \{1, \ldots, T\} \) **do**

- Compute stochastic gradient: \( \nabla_{\theta,t} = \nabla_{\theta} \ell_t(\theta) \)
- Update biased estimate of first moment: \( m_t = \beta_1 m_{t-1} + (1 - \beta_1) \cdot \nabla_{\theta,t} \)
- Update biased estimate of second moment: \( v_t = \beta_2 v_{t-1} + (1 - \beta_2) \cdot \nabla^2_{\theta,t} \)
- Compute bias corrected first moment: \( \hat{m}_t = m_t / (1 - \beta_1^t) \)
- Compute bias corrected second moment: \( \hat{v}_t = v_t / (1 - \beta_2^t) \)
- Perform optimistic gradient step: \( \theta_t = \theta_{t-1} - 2\eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \eta \frac{\hat{m}_{t-1}}{\sqrt{\hat{v}_{t-1}} + \epsilon} \)

**Return** \( \theta_T \)
Optimistic ADAM
Overview

1. What is “adversarial”?

2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
Boundary equilibrium GAN (BEGAN)

Samples

Interpolations of real images
Overview

1. What is “adversarial”?

2. Generative adversarial networks (GANs)
   - Vanilla GAN
   - WGAN

3. Paper review: Daskalakis, Training GANs with optimism, 2017

4. Boundary equilibrium GAN (BEGAN)

5. Adversarial examples
Adversarial Examples

- Examples that are similar to examples in the true distribution, but that fool a classifier [Szegedy, 2013]
- A demonstration of adversarial example [Goodfellow, 2014a]

Adversarial Examples

Art?
Adversarial Examples

- Why small changes?
- Universal, robust, targeted adversarial image patches in the real world [Brown, 2017]
- https://youtu.be/i1sp4X57TL4
Intriguing properties of neural networks.

Explaining and harnessing adversarial examples.

Adversarial patch.

Generative adversarial nets.
Advances in neural information processing systems, 2672–2680.


References V

avec les Memoires de Mathematique et de Physique. 1781.
Memoire sur la theorie des deblais et des remblais.
_Histoire de l’Academie Royale des Science, Annee 1781._

Optimal transport: old and new.
_Springer Science & Business Media._

Optimization, learning, and games with predictable sequences.
_Advances in Neural Information Processing Systems, 3066–3074._

Adam: A method for stochastic optimization.

Thank you!