Introduction to GANs pt. 2

a.k.a. «what could go wrong?»

N. Popov, M. Ryabinin

Faculty of Computer Science
Higher School of Economics

Jan. 26, 2018

Table of Contents

GANs should work

> I am coming to the conclusion that it's less about truly solving a 2 player game [...] and more about weaponizing a form of human calibrated overfitting
>
> some guy from reddit

- Two-player game;
- Optimal state — neither G nor D can improve;
- Our objective — reach that state;
- Just use gradient ascent, whatever; surely it will work out well?

GANs should work

So why don't they?

> I am coming to the conclusion that it's less about truly solving a 2 player game [...] and more about weaponizing a form of human calibrated overfitting
>
> — some guy from reddit

- Two-player game;
  Each maximizes their own objective
- Optimal state — neither G nor D can improve;
  In game theory this is called *Nash equilibrium*
- Our objective — reach that state;
  Gradient ascent *is not guaranteed* to reach Nash equilibria
- Just use gradient ascent, whatever; surely it will work out well?
  Nope.

Finding Nash equilibria by gradient descent

- It looks like GA used in GANs is a special case of regular GD;
- It's not, it is a generalization.
- Regular GA update:

$$x_{t+1} = x_t + hv(x_t);\ v(x) = \frac{\partial}{\partial x} f(x)$$

- Simultaneous GA update:

$$x_{t+1} = x_t + hv'(x_t);\ x = \begin{pmatrix} \theta \\ \phi \end{pmatrix};\ v'(x) = \begin{pmatrix} \frac{\partial}{\partial \theta} f(\theta, \phi) \\ \frac{\partial}{\partial \phi} g(\theta, \phi) \end{pmatrix}$$

Finding Nash equilibria by gradient descent

- It looks like GA used in GANs is a special case of regular GD;
- It's not, it is a generalization.
- Regular GA update:

$$x_{t+1} = x_t + hv(x_t);\ v(x) = \frac{\partial}{\partial x} f(x)$$

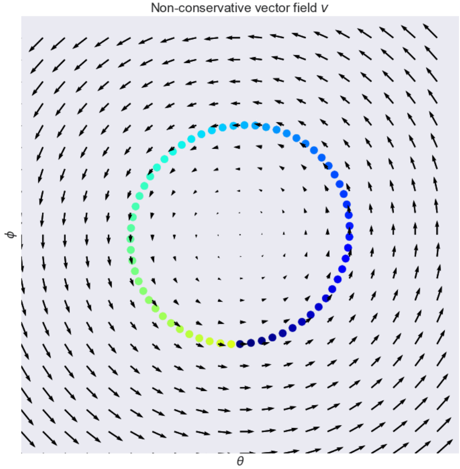
v is conservative

- Simultaneous GA update:

$$x_{t+1} = x_t + hv'(x_t);\ x = \begin{pmatrix} \theta \\ \phi \end{pmatrix};\ v'(x) = \begin{pmatrix} \frac{\partial}{\partial \theta} f(\theta, \phi) \\ \frac{\partial}{\partial \phi} g(\theta, \phi) \end{pmatrix}$$

v' may not be conservative

Non-conservative fields



This path does not look promising, does it?

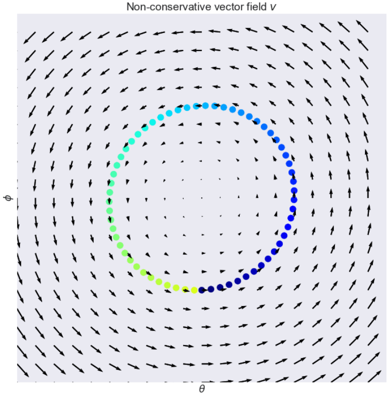
Possible solution

- **Mescheder et al. (2017)** propose a solution: construct a conservative field manually:

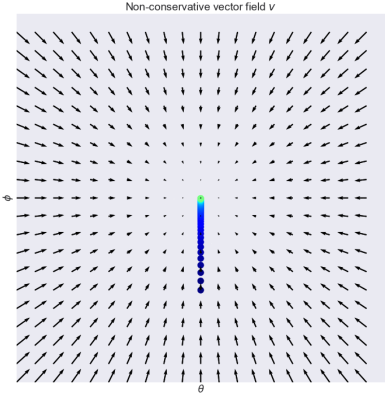
$$-\nabla L(x) = -\frac{\partial}{\partial x}\|v'(x)\|_2^2$$

- Basically, we want to minimize gradient norm;
- Fixed points are the same;
- At least GD converges now.

Comparison



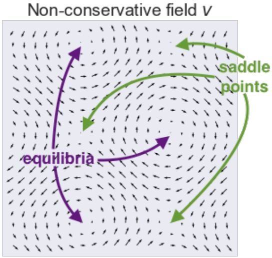
Non-conservative vector field v
ϕ
θ

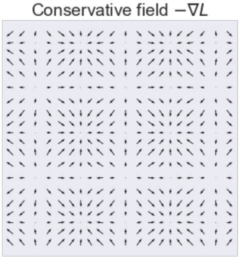


Non-conservative vector field v
ϕ
θ

Caveat

- All fixed points correspond to local minima now:

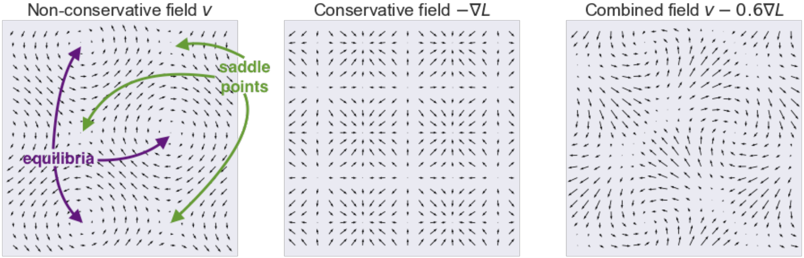




- We might converge to a saddle point, which is undesirable.

Caveat

- All fixed points correspond to local minima now:



- We might converge to a saddle point, which is undesirable.
- Let's combine both fields!
- This gives as a better behaved, but still non-conservative field.

Caveat

- How do we choose γ in $v - \gamma \nabla L$?
- If γ is too low, we still might not converge;
- If γ is too high, we might converge to a saddle point;
- Still an open problem.

Gradient Ascent is dead, long live Gradient Ascent!

- **Goodfellow et al. (2014)** propose a different training procedure:
- They propose optimizing G and D in turn;
- They suggest fully training D after each training step of G;
- They show that for powerful enough G and D this will converge (under some assumptions);
- These assumptions generally do not hold.

Caveat #2

- They assume that at each step of G it will improve it's quality;
- Very powerful D may prevent that by providing almost no usable gradient to G;
- The problem lies in the training criterion used by D:
- The original training criterion for D is as such:

$$\max_D V(G, D) = \max_D \int_x p_{\mathrm{r}}(x) \log(D(x)) + p_{\mathrm{g}}(x) \log(1 - D(x)) \mathrm{d}x$$

- Which can be shown to be equal to

$$-\log(4) + 2 \cdot \mathrm{JSD}(\mathbb{P}_{\mathrm{r}} \| \mathbb{P}_{\mathrm{g}})$$

The problem with Jensen-Shannon divergence

- JSD is a function of density ratio: $\frac{p_r}{p_g}$;
- If the distributions have (almost) no overlap it is zero/infinity everywhere;
- No usable gradients to speak of.
- We can fix this by forcing them to overlap (e.g. by adding noise);
- Not a very satisfying solution, feels like a hack.

Wasserstein GAN

- **Arjovsky et al. (2017)** propose a different objective;
- They show that using Wasserstein distance is a more sensible approach:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} \mathrm{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

- Shows how much "mass" has to be moved to transform $\mathbb{P}_r$ into $\mathbb{P}_g$
- However, it is intractable and cannot be computed directly.
- Kantorovich-Rubinstein duality:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq K} \left(\mathrm{E}_{x\sim P_r}[f(x)] - \mathrm{E}_{x\sim P_g}[f(x)]\right)$$

- Let's approximate supremum over Lipschitz functions with a constrained D.

Wasserstein GAN

This is somewhat different from original GAN:

- The discriminator no longer discriminates; hence, they propose the name “critic”;
- The outputs of the critic serve no purpose and are discarded;
- G is trained as usual, via critic’s gradients.
- Also, WGANs can be and were improved even further, deviating even more from a regular GAN.

GANs fixed?

Have we fixed everything wrong with GANs?

- You wish.
- There is a billion other, less fundamental problems
- Definitely a lot not yet discovered;
- Even with all these fixes, GANs are a pain to train, and the quality of results could definitely be improved;
- Still, a lot of progress is being made to mitigate this.

Generator – Discriminator disbalance

When one part significantly outperforms another, bad things happen:

- When D is more powerful than G, G can not improve at all.
- When G is exploiting D's weaknesses too well, D generally can't adapt neither.

We've already seen this problem addressed with WGANs; what other solutions are possible?

Generator – Discriminator disbalance

- Use noise in D;
  - Conceptually: hinders D's abilities, slows it down;
  - Mathematically: see above; makes distibutions overlap.
- Train only the weakest part:
  - e.g. train D to maximum, then train G (as in WGAN)
  - hard to measure fitness of one part
- Experience replay and other RL tricks
- Hyperparameter/random seed tuning

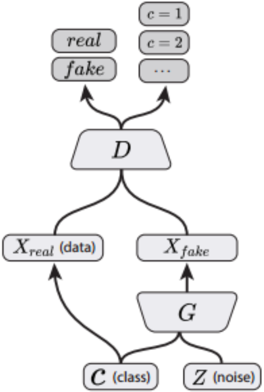
Mode collapse

Generator produces similar outputs for different inputs.

- Happens because there is no direct incentive to produce different images, as long as D is fooled
- Theoretically should not happen; ha-ha, theory.
- Solved by incentivizing variance:
  - Minibatch Discrimination: D can directly compare images in a batch
  - Unrolled GAN: prevents "cat-and-mousing".
  - Train several Gs for different modes (not recommended).
- May be caused by batch normalization.

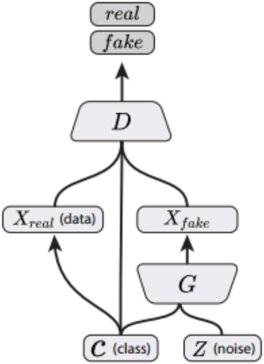
Use all available information

Auxillary classifier GAN



AC-GAN
(Present Work)

Conditional GAN



Conditional GAN
(Mirza & Osindero, 2014)

Useful hints

- Avoid sparse gradients:
  - ReLU $\rightarrow$ LeakyReLU
  - MaxPool2D $\rightarrow$ AvgPool2D / Conv2D + stride
- Regularization matters!
  - Sometimes it defines architectures;
  - Use noise in G as source of randomness;
- Just use a good architecture
  - DCGAN is a good start; WGAN, WGAN-GP, BEGAN, ProGAN...
  - If that's not an option, use a hybrid: e.g. GAN + VAE.

Summary

- GANs are broken in more than one way;
- They are difficult to train, sometimes unstable, and overall inconvenient;
- Despite this, they represent state-of-the-art in a lot of fields, and see no competition.
- GANs are being fixed in more than one way!
- Hopefully, in a few years' time we will see a significant amount of progress.

References I

Generative Adversarial Networks
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
arXiv:1406.2661 [stat.ML]

Improved Techniques for Training GANs
Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen
arXiv:1606.03498 [cs.LG]

Wasserstein GAN
Martin Arjovsky, Soumith Chintala, Léon Bottou
arXiv:1701.07875 [stat.ML]

The Numerics of GANs
Lars Mescheder, Sebastian Nowozin, Andreas Geiger
arXiv:1705.10461 [cs.LG]

References II

Conditional Generative Adversarial Nets
Mehdi Mirza, Simon Osindero
arXiv:1411.1784 [cs.LG]

Conditional Image Synthesis With Auxiliary Classifier GANs
Augustus Odena, Christopher Olah, Jonathon Shlens
arXiv:1610.09585 [stat.ML]

Instance Noise: A trick for stabilising GAN training
Casper Kaae Sønderby, Ferenc Huszár (2016)
http://www.inference.vc/instance-noise-a-trick-for-stabilising-gan-training/