Understanding Generative Adversarial Networks (GAN)

Group Meeting Discussion

Friday, March 30, 2018

一个关于GAN的感性例子

假如你是一名篮球运动员，你想在下次比赛中得到上场机会。

于是在每一次训练赛之后你跟教练进行沟通：

- 你：教练，我想打球
- 教练：（评估你的训练赛表现之后）... 算了吧
- （你通过跟其他人比较，发现自己的运球很差，于是你苦练了一段时间）

- 你：教练，我想打球
- 教练：... 嗯 还不行
- （你发现大家投篮都很准，于是你苦练了一段时间的投篮）

- 你：教练，我想打球
- 教练：... 嗯 还有所欠缺
- （你发现你的身体不够壮，被人一碰就倒，于是你去泡健身房）

......

通过这样不断的努力和被拒绝，你最终在某一次训练赛之后得到教练的赞赏，获得了上场的机会。

值得一提的是在这个过程中，所有的候选球员都在不断地进步和提升。因而教练也要不断地通过对比场上球员和候补球员来学习分辨哪些球员是真正可以上场的，并且要“观察”得比球员更频繁。随着大家的成长教练也会会变得越来越严格。

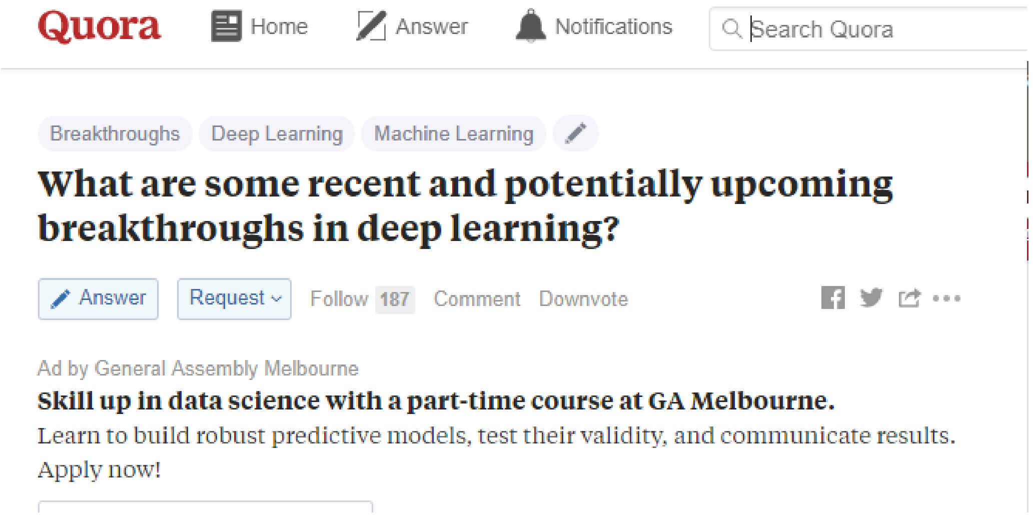
Background

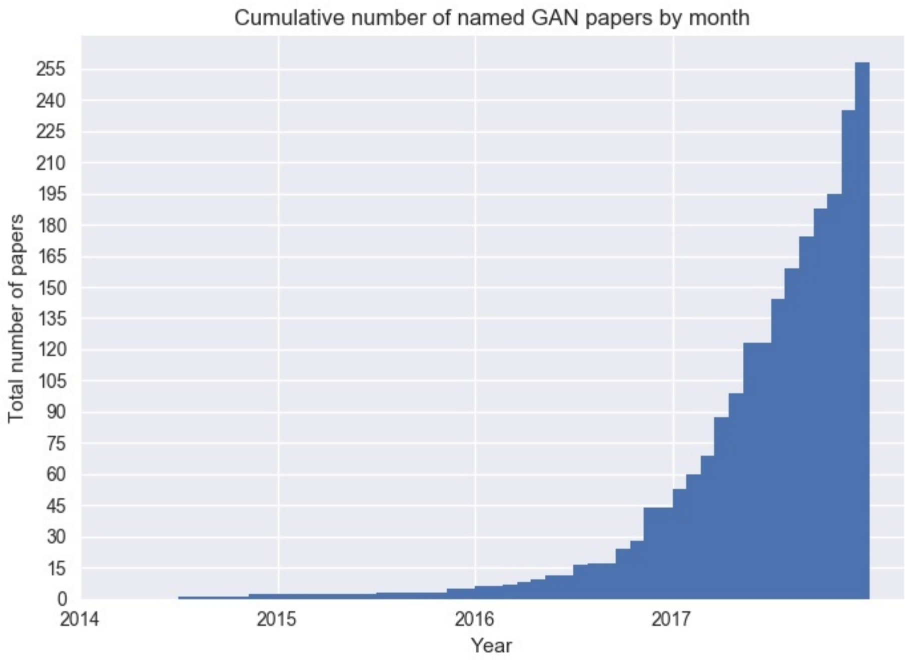
- Generative adversarial networks (GANs) are (deep) neural net architectures comprised of two nets, pitting one against the other (thus the "adversarial").

- GANs were introduced in a paper by Ian Goodfellow and other researchers at the University of Montreal, including Yoshua Bengio, in 2014.

- Referring to GANs, Facebook's AI research director Yann LeCun called adversarial training "the most interesting idea in the last 10 years in ML."

- GANs' potential is huge, because they can learn to mimic any distribution of data. The number of papers about GAN is growing fast.





- MIT Technology Review: The GANfather: The man who’s given machines the gift of imagination .

11

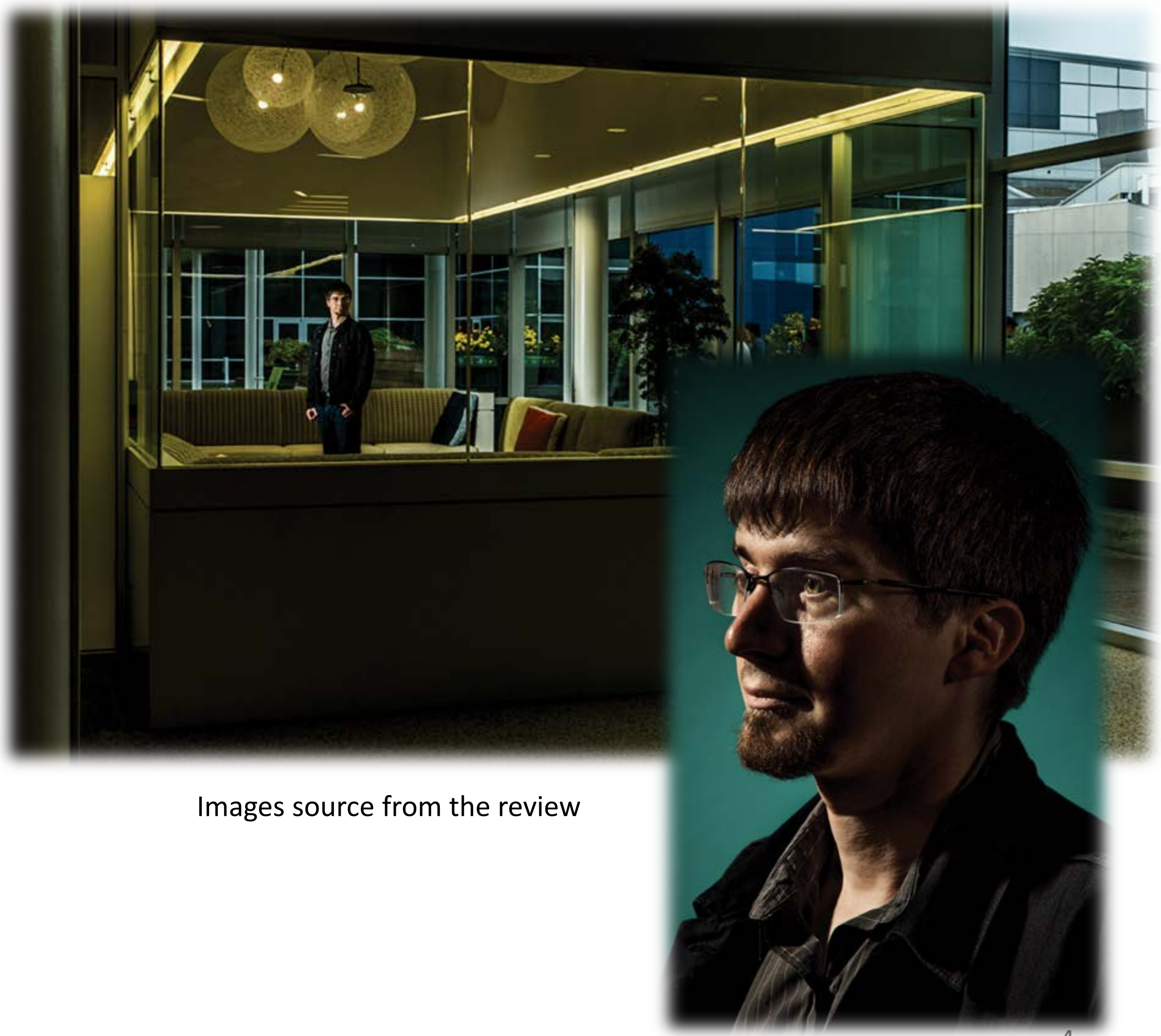
Intelligent Machines

The GANfather: The man who’s given machines the gift of imagination

By pitting neural networks against one another, Ian Goodfellow has created a powerful AI tool. Now he, and the rest of us, must face the consequences.

by Martin Giles February 21, 2018

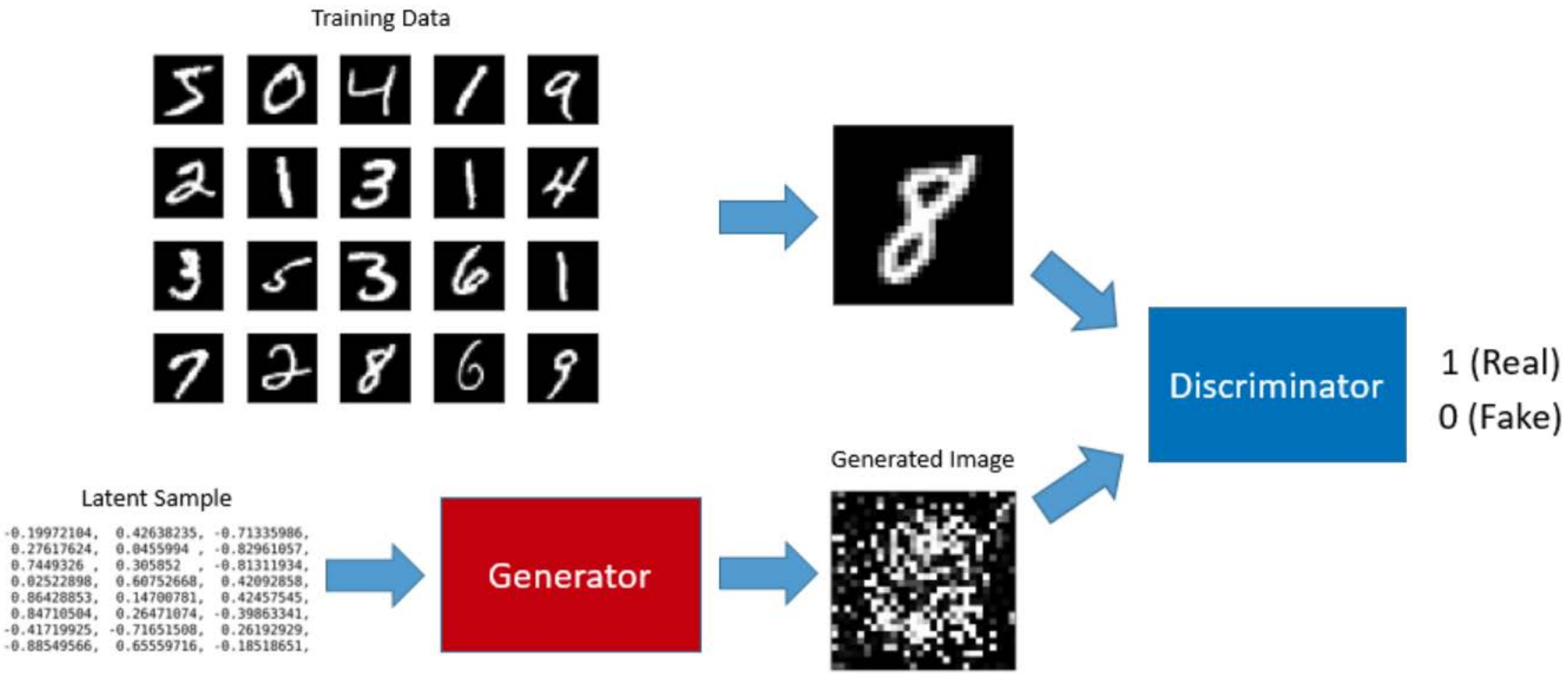
One night in 2014, Ian Goodfellow went drinking to celebrate with a fellow doctoral student who had just graduated. At Les 3 Brasseurs (The Three Brewers), a favorite Montreal watering hole, some friends asked for his help with a thorny project they were working on: a computer that could create photos by itself.



Images source from the review

Understanding GAN Framework

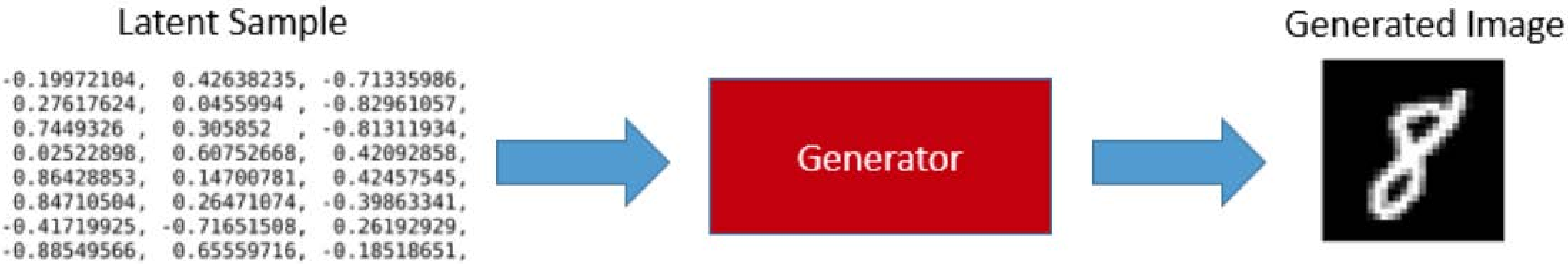
- A GAN can be trained to generate images from random noises. For example, we can train a GAN to generate digit images that look like hand-written digit images from MNIST database.
- A GAN has two parts in it: the generator that generates images and the discriminator that classifies real and fake images.



[Image source: Understanding Generative Adversarial Networks](#).

GAN Framework - Generator

- The input to the generator is a series of randomly generated numbers called latent sample. Once trained, the generator can produce digit images from latent samples.

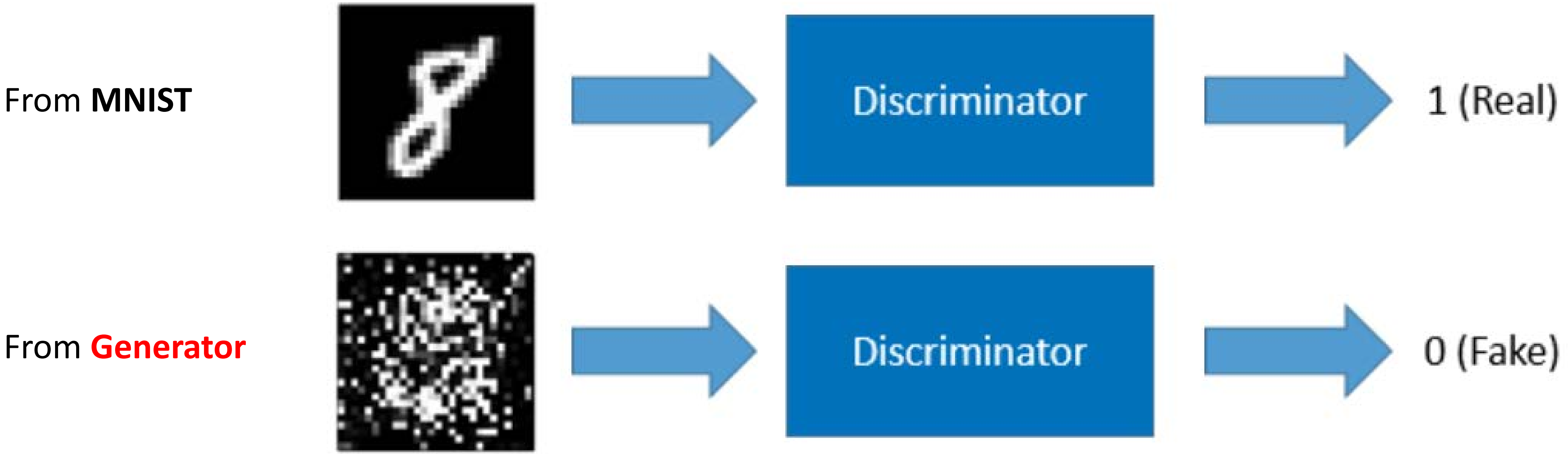


Our generator is a simple fully connected network that takes a latent sample (100 randomly generated numbers) and produces 784 data points which can be reshaped into a 28 x 28 digit image which is the size used by all MNIST digit images.
To train the generator, we need to train a GAN. Before talking about GAN, we shall discuss the discriminator.

[Image source: Understanding Generative Adversarial Networks](#).

GAN Framework - Discriminator

- The discriminator is a classifier trained using the supervised learning. It classifies whether an image is real (1) or not (0).

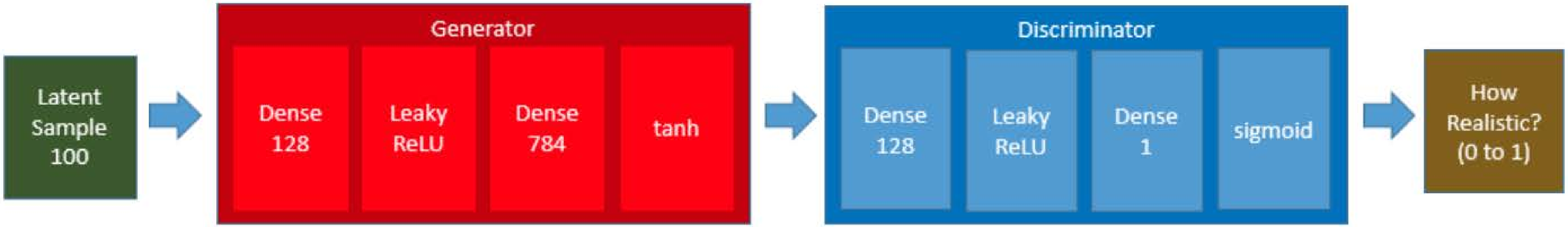


We train the discriminator using both the MNIST images and the images generated by the generator. If the input image is from the MNIST database, the discriminator should classify it as real. If the input image is from the generator, the discriminator should classify it as fake.

[Image source: Understanding Generative Adversarial Networks](#).

Training the GAN means Training the Generator

- We connect the generator and the discriminator to produce a GAN.



When we feed a latent sample to the GAN, the generator internally produces a digit image which is then passed to the discriminator for classification. If the generator does a good job, the discriminator returns a value close to 1 (high probability of the image being real).

We feed latent samples to the GAN while setting the expected outcome (label) to 1 (real) as we expect the generator to produce realistic image, and we expect the discriminator to say it is real or close to real.

However, the generator initially produces garbage images, and the loss value is high. So, the back-propagation updates the generator's weights to produce more realistic images as the training continues.

Image source: Understanding Generative Adversarial Networks.

How to train a GAN

- There is one catch in this process of training the generator via the GAN. We do not want the discriminator's weights to be affected because we are using the discriminator as merely a classifier.
- For this reason, we set the discriminator non-trainable during the generator training.

Image source: Understanding Generative Adversarial Networks.

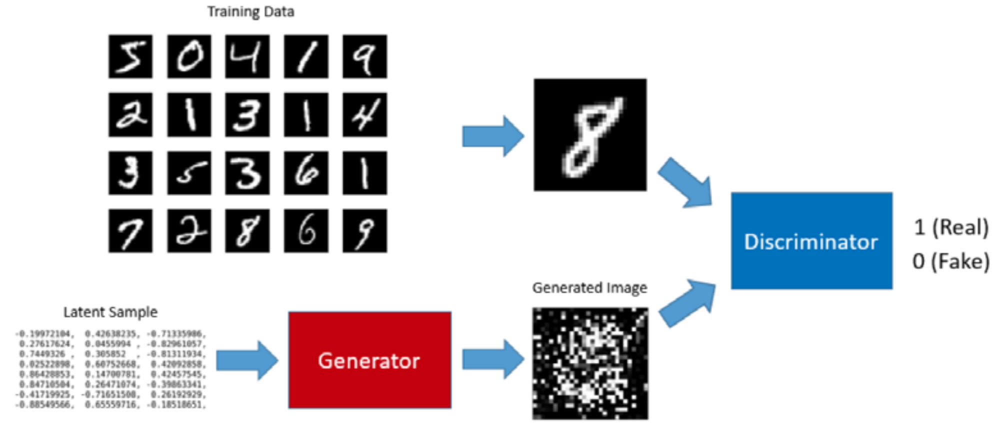
Training Loop of a GAN

- To get a better generator, we also need to train the discriminator as well so that it can do a good job as a classifier of real and fake images. We train the discriminator and the generator in turn in a loop as follows:

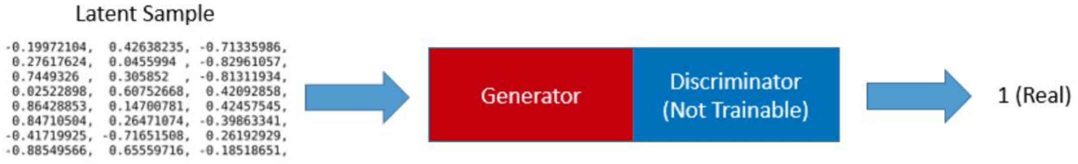
➢ Step 1) Set the discriminator trainable;
➢ Step 2) Train the discriminator with the real MNIST digit images and the images generated by the generator to classify the real and fake images.

➢ Step 3) Set the discriminator non-trainable;
➢ Step 4) Train the generator as part of the GAN. We feed latent samples into the GAN and let the generator to produce digit images and use the discriminator to classify the image.

The loop should ideally continue until they are both trained well and can not be improved any further.



1. The Discriminator Training



2. The Generator Training (via GAN)

[Image source: Understanding Generative Adversarial Networks](#).

Training Loop of a GAN

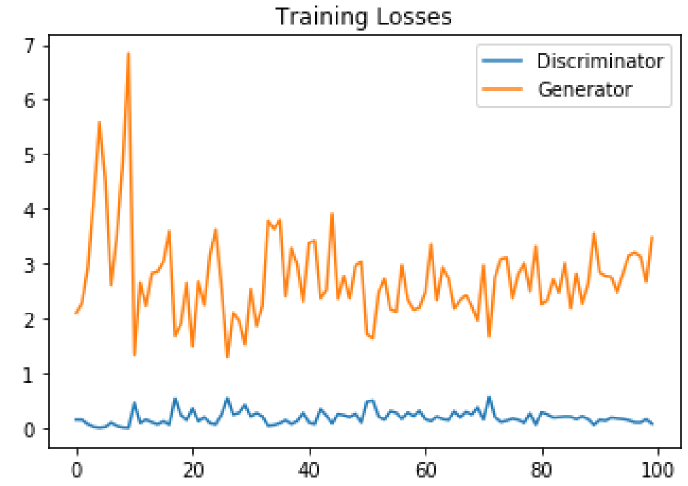
- Prepare a batch of real images
- Prepare a batch of fake images generated by the generator using latent samples
- Make the discriminator trainable
- Train the discriminator to classify the real and fake images
- Make the discriminator non-trainable
- Train the generator via the GAN

Stabilizing GAN

As it turns out, training a GAN is quite hard, and there are many tricks and heuristics required. It is because the discriminator and the generator are not cooperating and individually learning to predict better.

For example, the generator might learn to fool the discriminator with garbage. Ideally, the discriminator should learn earlier than the generator so that it can classify images accurately.

Two different training results



[Image source: Understanding Generative Adversarial Networks](#).

But does it actually work?

- The result of the simple GAN is not outstanding. Some of them look pretty good but others are not.

As it turns out, training a GAN requires lots of hacks as per **How to Train a GAN? Tips and tricks to make GANs work** such as label smoothing and other techniques.

"There are all sorts of empirical quirks. If I train the discriminator much faster than the generator, the generator gives up learning. In some case, the generator learns to deceive the discriminator and makes the discriminator unable to learn to classify properly."

-- Naoki Shibuya

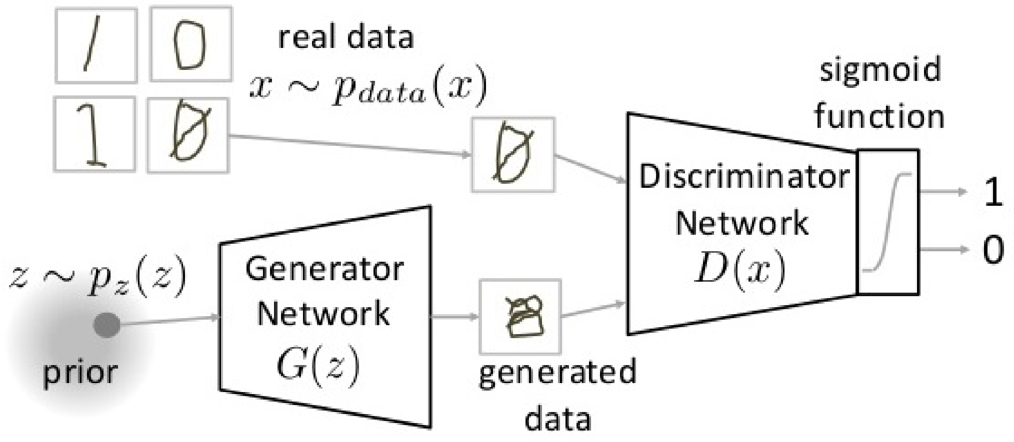
The results are not outstanding as we are using simple networks. Deep Convolutional GAN (aka DCGAN) would produce better results than this.

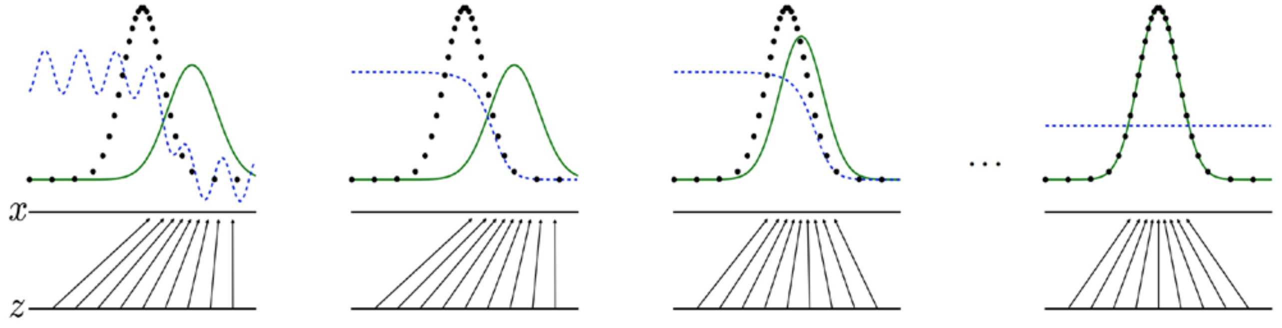
Testing results from **Generator**

Image source: Understanding Generative Adversarial Networks.

The Loss Function of a GAN

- G(z): Generator Network
- D(x): Discriminator Network
- **Pdata(x): the distribution of real data;**
- **X: sample from Pdata(x).**
- **P(z): distribution of generator;**
- **Z: sample from Pz(z).**
- **D(x): discriminative distribution**





(a) (b) (c) (d)

a) an adversarial pair near convergence.
b) D is trained to discriminate samples from data.
c) After an update to G.
d) After several steps of training.

if G and D have enough capacity, they will reach a point at which both cannot improve.
The discriminator is unable to differentiate between the two distributions, i.e. D(x) = ½.

Image from : https://www.analyticsvidhya.com/blog/2017/06/introductory-generative-adversarial-networks-gans/ ; GAN paper.

The Loss Function of a GAN

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

- Cross entropy: $H(p, q) := -\sum_i p_i \log q_i$ where p and q denote a "true" and an "empirical/estimated" distribution.
- For one data point x_1 and its label from a binary classification task, we get the following loss function, where here I've changed the input to be more precise:

One samples: $$H((x_1, y_1), D) = -y_1 \log D(x_1) - (1 - y_1)\log(1 - D(x_1))$$

N samples: $$H((x_i, y_i)_{i=1}^N, D) = -\sum_{i=1}^N y_i \log D(x_i) - \sum_{i=1}^N (1 - y_i)\log(1 - D(x_i))$$

Infinite samples (yi = 1/2): $$H((x_i, y_i)_{i=1}^\infty, D) = -\frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}}\Big[\log D(x)\Big] - \frac{1}{2}\mathbb{E}_z\Big[\log(1 - D(G(z)))\Big]$$

The Loss Function of a GAN (Cont.)

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

- Understanding the $\min_G \max_D V(D, G)$.

- V(D,G) means the measurement of difference of x and z.
- $\max_D V(D, G)$ means fix the G and train the D.
- $\min_G L$ $(where\ L = \max_D V(D, G)$) means fix D to train G. Minimize the different between x and z.

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$\begin{aligned} V(G, D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \to a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$, concluding the proof. □

Proposition 2. *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*

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

then p_g converges to p_{data}

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha \in \mathcal{A}} f_\alpha(x)$. This is equivalent to computing a gradient descent update for p_g at the optimal D given the corresponding G. $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima as proven in Thm 1, therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. □

The Training Algorithm (GAN Paper)

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:

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

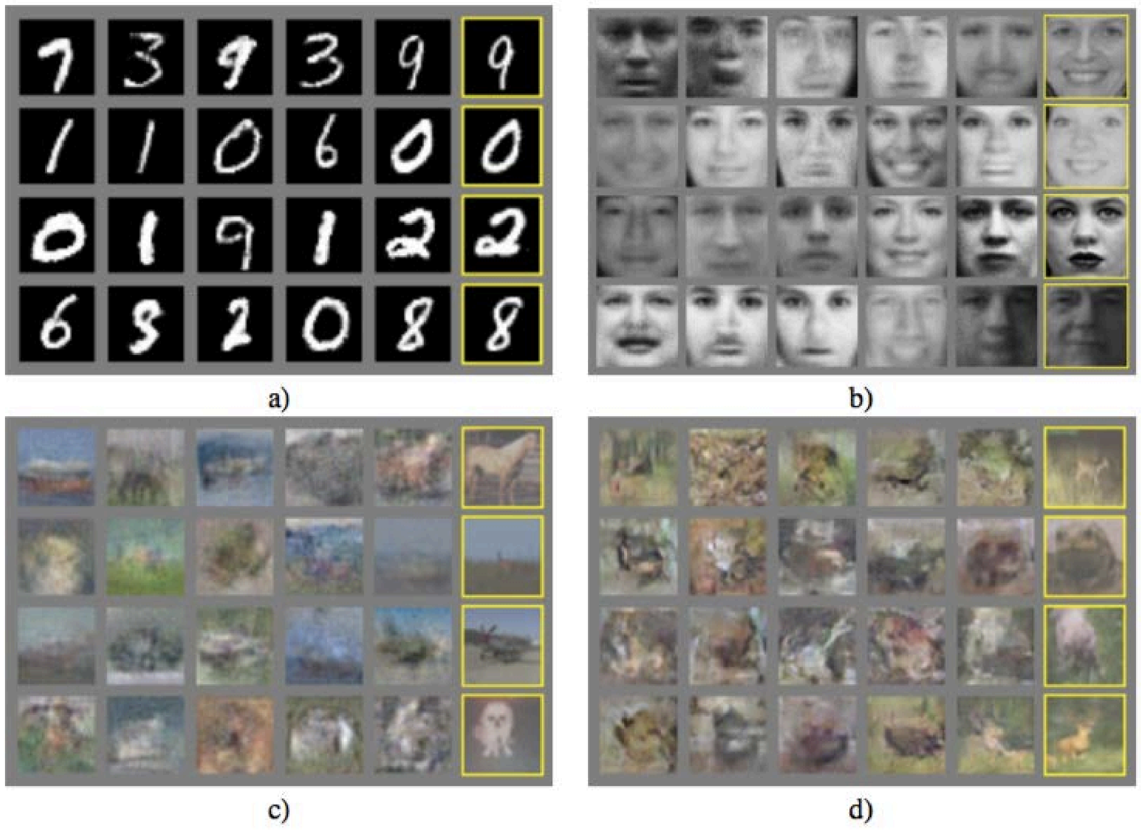
end for

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

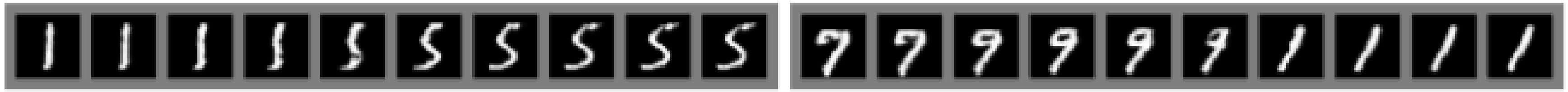
Samples drawn from the generator net after training

Experimental Results from the GAN paper by Ian Goodfellow, etc.

- a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)



- Digits obtained by linearly interpolating between coordinates in z space of the full model.



Images from the GAN paper.

More Research and Application

Cornell University Library

We gratefully acknowledg
the Simo
and mem

arXiv.org > stat > arXiv:1701.04862

Search or Article ID | All pap

(Help | Advanced search)

Statistics > Machine Learning

Towards Principled Methods for Training Generative Adversarial Networks

Martin Arjovsky, Léon Bottou

https://arxiv.org/abs/1701.04862

(Submitted on 17 Jan 2017)

The goal of this paper is not to introduce a single algorithm or method, but to make theoretical steps towards fully understanding the training dynamics of generative adversarial networks. In

Download:
- PDF
- Other formats

(license)

Current browse c
stat.ML
< prev | next >
new | recent | 1701

Change to brows

Cornell University Library

We gratefully acknowledge suppo
the Simons Foun
and member instit

arXiv.org > stat > arXiv:1701.07875

Search or Article ID | All papers

(Help | Advanced search)

Statistics > Machine Learning

Wasserstein GAN

Martin Arjovsky, Soumith Chintala, Léon Bottou

(Submitted on 26 Jan 2017 (v1), last revised 6 Dec 2017 (this version, v3))

We introduce a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, we show that we can improve the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searches. Furthermore, we show that the corresponding optimization problem is sound, and provide extensive theoretical work highlighting the deep connections to other distances between distributions.

Subjects: Machine Learning (stat.ML); Learning (cs.LG)
Cite as: arXiv:1701.07875 [stat.ML]
(or arXiv:1701.07875v3 [stat.ML] for this version)

https://arxiv.org/pdf/1701.07875.pdf

Submission history

From: Martin Arjovsky [view email]

Download:
- PDF
- Other formats

(license)

Current browse context:
stat.ML
< prev | next >
new | recent | 1701

Change to browse by:
cs
cs.LG
stat

References & Citations
- NASA ADS

Bookmark (what is this?)

reddit MACHINELEARNING comments other discussions (1)

149 [R] [1701.07875] Wasserstein GAN (arxiv.org)
submitted 1 year ago by ajmooch
167 comments share save hide give gold report crosspost

sorted by: best

you are viewing a single comment's thread.
view the rest of the comments →

[–] ian_goodfellow Google Brain 26 points 1 year ago

"mode collapse comes from the fact that the optimal generator for a fixed discriminator is a sum of deltas on the points the discriminator assigns the highest values, as brilliantly observed by [11]"

This was actually known since the first GAN paper. I don't think [11] claim identifying this as a contribution. Their solution to that problem is very nice though.

You can see this claim in my slides, for example these: http://www.iangoodfellow.com/slides/2016-08-31-Berkeley.pdf "Fully optimizing the generator with the discriminator held constant results in mapping all points to the argmax of the discriminator"

It's worth mentioning that, depending on the structure of the discriminator, the set of points defining the argmax might not be isolated deltas, so the description in this paper isn't quite correct.

Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

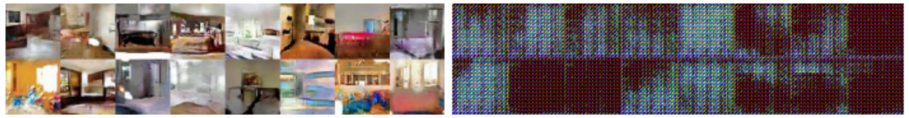


Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [18]). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. Left: WGAN algorithm. Right: standard GAN formulation. As we can see the standard GAN failed to learn while the WGAN still was able to produce samples.

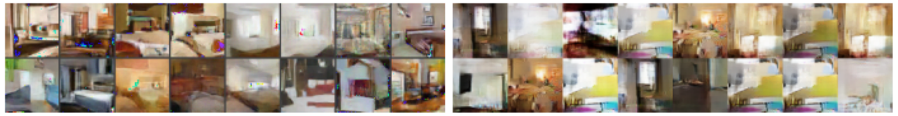
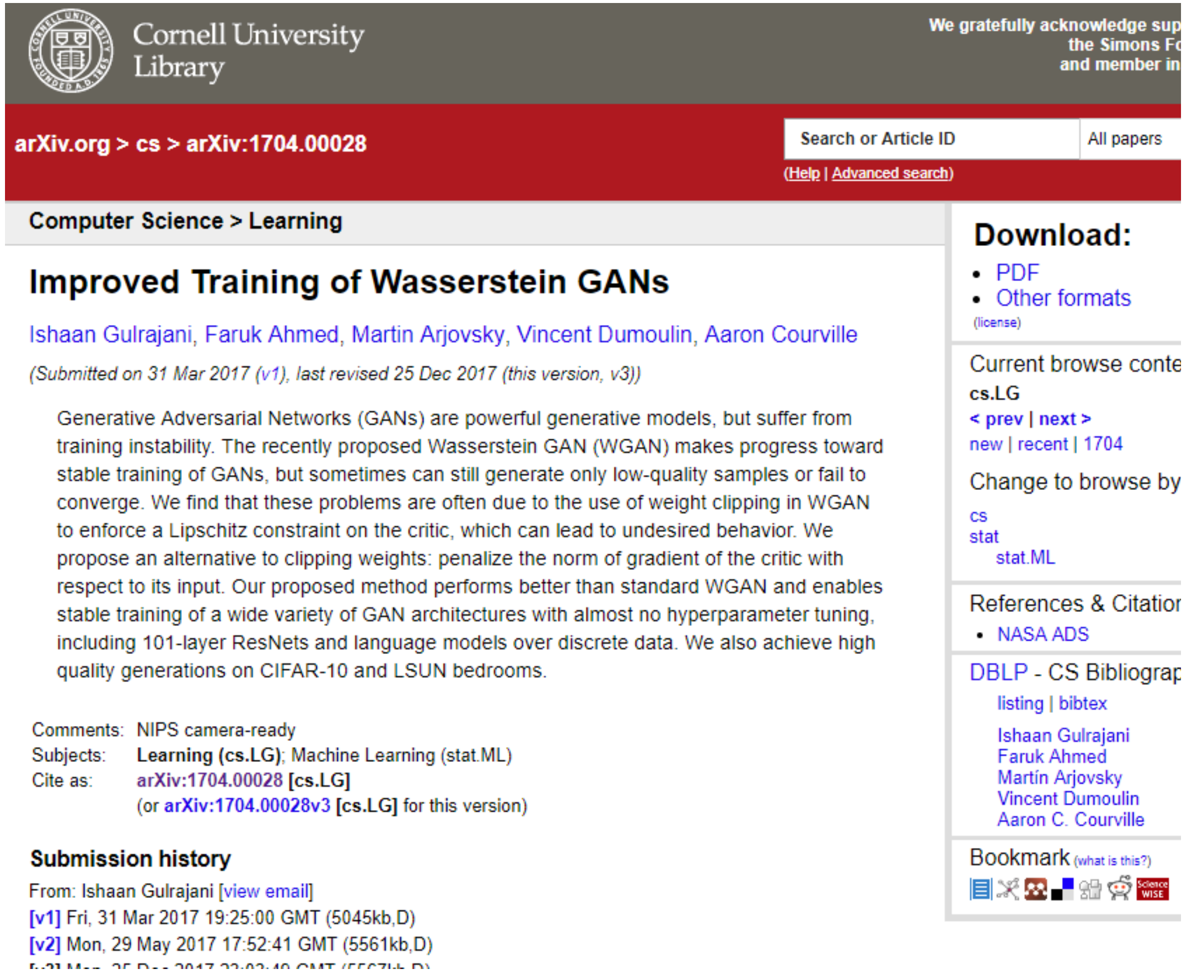


Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was able to produce samples, lower quality than the DCGAN, and of higher quality than the MLP of the standard GAN. Note the significant degree of mode collapse in the GAN MLP.



Cornell University Library

We gratefully acknowledge sup
the Simons F
and member in

arXiv.org > cs > arXiv:1704.00028

Search or Article ID | All papers

(Help | Advanced search)

Computer Science > Learning

Improved Training of Wasserstein GANs

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville

(Submitted on 31 Mar 2017 (v1), last revised 25 Dec 2017 (this version, v3))

Generative Adversarial Networks (GANs) are powerful generative models, but suffer from training instability. The recently proposed Wasserstein GAN (WGAN) makes progress toward stable training of GANs, but sometimes can still generate only low-quality samples or fail to converge. We find that these problems are often due to the use of weight clipping in WGAN to enforce a Lipschitz constraint on the critic, which can lead to undesired behavior. We propose an alternative to clipping weights: penalize the norm of gradient of the critic with respect to its input. Our proposed method performs better than standard WGAN and enables stable training of a wide variety of GAN architectures with almost no hyperparameter tuning, including 101-layer ResNets and language models over discrete data. We also achieve high quality generations on CIFAR-10 and LSUN bedrooms.

Comments: NIPS camera-ready
Subjects: **Learning (cs.LG)**; Machine Learning (stat.ML)
Cite as: **arXiv:1704.00028 [cs.LG]**
(or **arXiv:1704.00028v3 [cs.LG]** for this version)

Submission history

From: Ishaan Gulrajani [view email]
[v1] Fri, 31 Mar 2017 19:25:00 GMT (5045kb,D)
[v2] Mon, 29 May 2017 17:52:41 GMT (5561kb,D)

Download:
- PDF
- Other formats

(license)

Current browse conte
cs.LG
< prev | next >
new | recent | 1704

Change to browse by
cs
stat
stat.ML

References & Citatior
- NASA ADS

DBLP - CS Bibliograp
listing | bibtex
Ishaan Gulrajani
Faruk Ahmed
Martín Arjovsky
Vincent Dumoulin
Aaron C. Courville

Bookmark (what is this?)

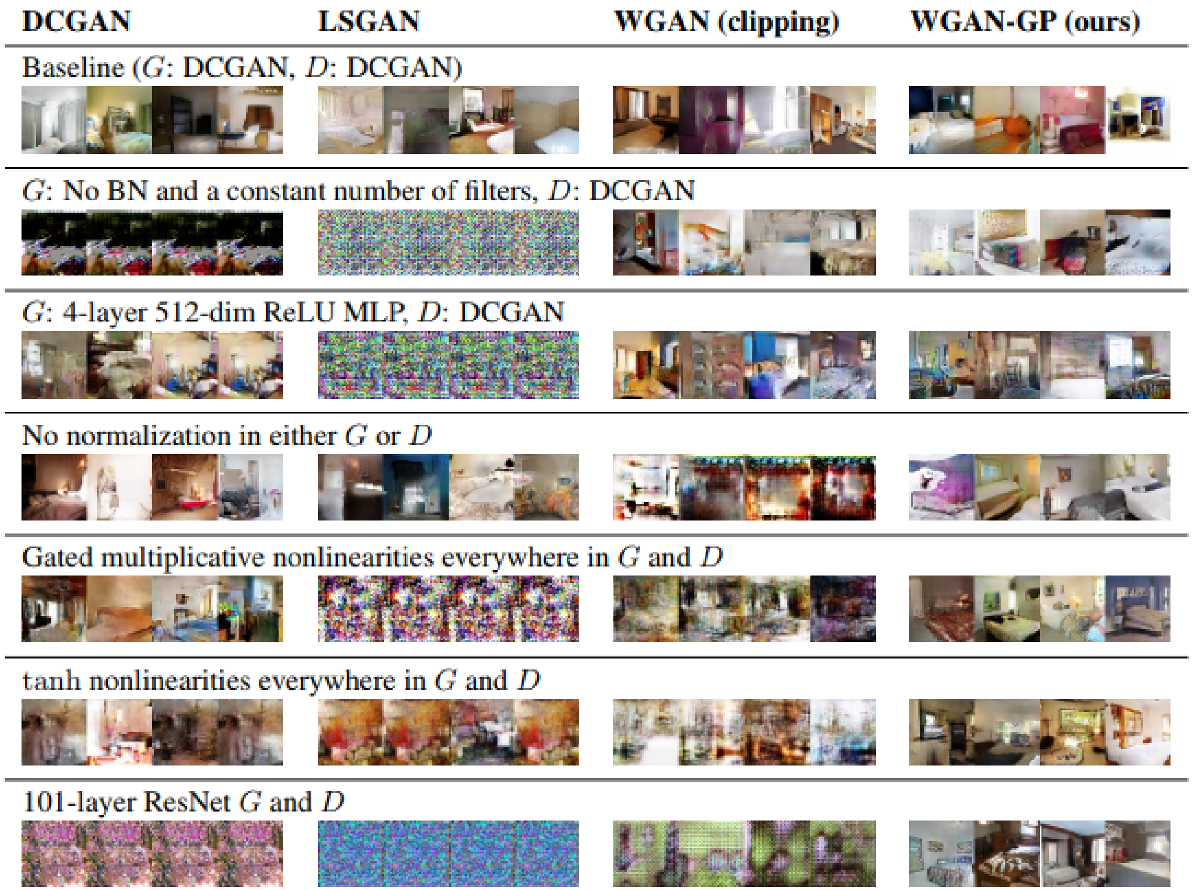


Figure 2: Different GAN architectures trained with different methods. We only succeeded in training every architecture with a shared set of hyperparameters using WGAN-GP.

https://arxiv.org/pdf/1704.00028.pdf

GAN ZOO (A-E)

- 3D-ED-GAN — Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN — Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN — Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-RecGAN — 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN — ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks(github)
- ABC-GAN — GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN — Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN — Face Aging With Conditional Generative Adversarial Networks
- ACtuAL — ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN — AdaGAN: Boosting Generative Models
- AdvGAN — Generating adversarial examples with adversarial networks
- AE-GAN — AE-GAN: adversarial eliminating with GAN
- AEGAN — Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN — Amortised MAP Inference for Image Super-resolution
- AL-CGAN — Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI — Adversarially Learned Inference
- AlignGAN — AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
- AM-GAN — Activation Maximization Generative Adversarial Nets
- AnoGAN — Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- APE-GAN — APE-GAN: Adversarial Perturbation Elimination with GAN
- ARAE — Adversarially Regularized Autoencoders for Generating Discrete Structures (github)
- ARDA — Adversarial Representation Learning for Domain Adaptation
- ARIGAN — ARIGAN: Synthetic Arabidopsis Plants using Generative Adversarial Network
- ArtGAN — ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- AttGAN — Arbitrary Facial Attribute Editing: Only Change What You Want
- AttnGAN — AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks
- b-GAN — Generative Adversarial Nets from a Density Ratio Estimation Perspective
- Bayesian GAN — Deep and Hierarchical Implicit Models (github)
- Bayesian GAN — Bayesian GAN
- BCGAN — Bayesian Conditional Generative Adverserial Networks
- BCGAN — Bidirectional Conditional Generative Adversarial networks
- BEGAN — BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BGAN — Binary Generative Adversarial Networks for Image Retrieval (github)
- BicycleGAN — Toward Multimodal Image-to-Image Translation (github)
- BiGAN — Adversarial Feature Learning
- BS-GAN — Boundary-Seeking Generative Adversarial Networks
- C-GAN — Face Aging with Contextual Generative Adversarial Nets
- C-RNN-GAN — C-RNN-GAN: Continuous recurrent neural networks with adversarial training (github)
- CA-GAN — Composition-aided Sketch-realistic Portrait Generation
- CaloGAN — CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks (github)
- CAN — CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms
- CapsuleGAN — CapsuleGAN: Generative Adversarial Capsule Network
- CatGAN — Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CatGAN — CatGAN: Coupled Adversarial Transfer for Domain Generation
- CausalGAN — CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training
- CC-GAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks (github)
- CDcGAN — Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network
- CFG-GAN — Composite Functional Gradient Learning of Generative Adversarial Models
- CGAN — Conditional Generative Adversarial Nets
- CGAN — Controllable Generative Adversarial Network
- Chekhov GAN — An Online Learning Approach to Generative Adversarial Networks
- CipherGAN — Unsupervised Cipher Cracking Using Discrete GANs
- CM-GAN — CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning
- CoAtt-GAN — Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning
- CoGAN — Coupled Generative Adversarial Networks
- ComboGAN — ComboGAN: Unrestrained Scalability for Image Domain Translation (github)
- ConceptGAN — Learning Compositional Visual Concepts with Mutual Consistency
- Conditional cycleGAN — Conditional CycleGAN for Attribute Guided Face Image Generation
- constrast-GAN — Generative Semantic Manipulation with Contrasting GAN
- Context-RNN-GAN — Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- Coulomb GAN — Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields
- Cover-GAN — Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks
- Cramèr GAN — The Cramer Distance as a Solution to Biased Wasserstein Gradients
- Cross-GAN — Crossing Generative Adversarial Networks for Cross-View Person Re-identification
- crVAE-GAN — Channel-Recurrent Variational Autoencoders
- CS-GAN — Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN — CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN — Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (github)
- D-GAN — Differential Generative Adversarial Networks: Synthesizing Non-linear Facial Variations with Limited Number of Training Data
- D2GAN — Dual Discriminator Generative Adversarial Nets
- DA-GAN — DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks (with Supplementary Materials)
- DAGAN — Data Augmentation Generative Adversarial Networks
- DAN — Distributional Adversarial Networks
- DCGAN — Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (github)
- DeblurGAN — DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks (github)
- Defense-GAN — Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models
- DeliGAN — DeLiGAN : Generative Adversarial Networks for Diverse and Limited Data (github)
- DF-GAN — Learning Disentangling and Fusing Networks for Face Completion Under Structured Occlusions
- DiscoGAN — Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DistanceGAN — One-Sided Unsupervised Domain Mapping
- DM-GAN — Dual Motion GAN for Future-Flow Embedded Video Prediction
- DNA-GAN — DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images
- dp-GAN — Differentially Private Releasing via Deep Generative Model
- DP-GAN — DP-GAN: Diversity-Promoting Generative Adversarial Network for Generating Informative and Diversified Text
- DPGAN — Differentially Private Generative Adversarial Network http://arxiv.org/abs/1802.06739 (github)
- DR-GAN — Representation Learning by Rotating Your Faces
- DRAGAN — How to Train Your DRAGAN (github)
- DRPAN — Discriminative Region Proposal Adversarial Networks for High-Quality Image-to-Image Translation
- DSP-GAN — Depth Structure Preserving Scene Image Generation
- DTN — Unsupervised Cross-Domain Image Generation
- DualGAN — DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- Dualing GAN — Dualing GANs
- Dynamics Transfer GAN — Dynamics Transfer GAN: Generating Video by Transferring Arbitrary Temporal Dynamics from a Source Video to a Single Target Image
- EBGAN — Energy-based Generative Adversarial Network
- ecGAN — eCommerceGAN : A Generative Adversarial Network for E-commerce
- ED//GAN — Stabilizing Training of Generative Adversarial Networks through Regularization
- EGAN — Enhanced Experience Replay Generation for Efficient Reinforcement Learning
- EnergyWGAN — Energy-relaxed Wassertein GANs (EnergyWGAN): Towards More Stable and High Resolution Image Generation
- ExGAN — Eye In-Painting with Exemplar Generative Adversarial Networks
- ExprGAN — ExprGAN: Facial Expression Editing with Controllable Expression Intensity

- f-CLSWGAN — Feature Generating Networks for Zero-Shot Learning
- f-GAN — f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN — Towards Large-Pose Face Frontalization in the Wild
- FIGAN — Frame Interpolation with Multi-Scale Deep Loss Functions and Generative Adversarial Networks
- Fila-GAN — Synthesizing Filamentary Structured Images with GANs
- First Order GAN — First Order Generative Adversarial Networks
- Fisher GAN — Fisher GAN
- Flow-GAN — Flow-GAN: Bridging implicit and prescribed learning in generative models
- FSEGAN — Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition
- FTGAN — Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture
- FusedGAN — Semi-supervised FusedGAN for Conditional Image Generation
- FusionGAN — Learning to Fuse Music Genres with Generative Adversarial Dual Learning
- G2-GAN — Geometry Guided Adversarial Facial Expression Synthesis
- GAGAN — GAGAN: Geometry-Aware Generative Adverserial Networks
- GAMN — Generative Adversarial Mapping Networks
- GAN — Generative Adversarial Networks (github)
- GAN-ATV — A Novel Approach to Artistic Textual Visualization via GAN
- GAN-CLS — Generative Adversarial Text to Image Synthesis (github)
- GAN-RS — Towards Qualitative Advancement of Underwater Machine Vision with Generative Adversarial Networks
- GAN-sep — GANs for Biological Image Synthesis (github)
- GAN-VFS — Generative Adversarial Network-based Synthesis of Visible Faces from Polarimetric Thermal Faces
- GANCS — Deep Generative Adversarial Networks for Compressed Sensing Automates MRI
- GANDI — Guiding the search in continuous state-action spaces by learning an action sampling distribution from off-target samples
- GANG — GANGs: Generative Adversarial Network Games
- GANosaic — GANosaic: Mosaic Creation with Generative Texture Manifolds
- GAP — Context-Aware Generative Adversarial Privacy
- GAWWN — Learning What and Where to Draw (github)
- GC-GAN — Geometry-Contrastive Generative Adversarial Network for Facial Expression Synthesis
- GeneGAN — GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data (github)
- GeoGAN — Generating Instance Segmentation Annotation by Geometry-guided GAN
- Geometric GAN — Geometric GAN
- GLCA-GAN — Global and Local Consistent Age Generative Adversarial Networks
- GMAN — Generative Multi-Adversarial Networks
- GMM-GAN — Towards Understanding the Dynamics of Generative Adversarial Networks
- GoGAN — Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN — GP-GAN: Towards Realistic High-Resolution Image Blending (github)
- GP-GAN — GP-GAN: Gender Preserving GAN for Synthesizing Faces from Landmarks
- GPU — A generative adversarial framework for positive-unlabeled classification
- GRAN — Generating images with recurrent adversarial networks (github)
- GraspGAN — Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping
- HAN — Chinese Typeface Transformation with Hierarchical Adversarial Network
- HP-GAN — HP-GAN: Probabilistic 3D human motion prediction via GAN
- HR-DCGAN — High-Resolution Deep Convolutional Generative Adversarial Networks
- IAN — Neural Photo Editing with Introspective Adversarial Networks (github)
- IcGAN — Invertible Conditional GANs for image editing (github)
- ID-CGAN — Image De-raining Using a Conditional Generative Adversarial Network
- IdCycleGAN — Face Translation between Images and Videos using Identity-aware CycleGAN
- IFcVAEGAN — Conditional Autoencoders with Adversarial Information Factorization
- iGAN — Generative Visual Manipulation on the Natural Image Manifold (github)
- Improved GAN — Improved Techniques for Training GANs (github)
- In2I — In2I : Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks
- InfoGAN — InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets(github)
- IRGAN — IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval models
- Iterative-GAN — Two Birds with One Stone: Iteratively Learn Facial Attributes with GANs (github)
- IVE-GAN — IVE-GAN: Invariant Encoding Generative Adversarial Networks
- iVGAN — Towards an Understanding of Our World by GANing Videos in the Wild (github)
- IWGAN — On Unifying Deep Generative Models
- KBGAN — KBGAN: Adversarial Learning for Knowledge Graph Embeddings
- KGAN — KGAN: How to Break The Minimax Game in GAN
- l-GAN — Representation Learning and Adversarial Generation of 3D Point Clouds
- LAC-GAN — Grounded Language Understanding for Manipulation Instructions Using GAN-Based Classification
- LAGAN — Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN — Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks (github)
- LB-GAN — Load Balanced GANs for Multi-view Face Image Synthesis
- LD-GAN — Linear Discriminant Generative Adversarial Networks
- LDAN — Label Denoising Adversarial Network (LDAN) for Inverse Lighting of Face Images
- LeakGAN — Long Text Generation via Adversarial Training with Leaked Information
- LeGAN — Likelihood Estimation for Generative Adversarial Networks
- LGAN — Global versus Localized Generative Adversarial Nets
- LR-GAN — LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation
- LS-GAN — Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities
- LSGAN — Least Squares Generative Adversarial Networks
- MAD-GAN — Multi-Agent Diverse Generative Adversarial Networks
- MAGAN — MAGAN: Margin Adaptation for Generative Adversarial Networks
- MalGAN — Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN
- MaliGAN — Maximum-Likelihood Augmented Discrete Generative Adversarial Networks
- manifold-WGAN — Manifold-valued Image Generation with Wasserstein Adversarial Networks
- MARTA-GAN — Deep Unsupervised Representation Learning for Remote Sensing Images
- MaskGAN — MaskGAN: Better Text Generation via Filling in the _______
- MC-GAN — Multi-Content GAN for Few-Shot Font Style Transfer (github)
- McGAN — McGan: Mean and Covariance Feature Matching GAN
- MD-GAN — Learning to Generate Time-Lapse Videos Using Multi-Stage Dynamic Generative Adversarial Networks
- MDGAN — Mode Regularized Generative Adversarial Networks
- MedGAN — Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks
- MGAN — Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks (github)
- MGGAN — Multi-Generator Generative Adversarial Nets
- MIL-GAN — Multimodal Storytelling via Generative Adversarial Imitation Learning
- MIX+GAN — Generalization and Equilibrium in Generative Adversarial Nets (GANs)
- MLGAN — Metric Learning-based Generative Adversarial Network
- MMD-GAN — MMD GAN: Towards Deeper Understanding of Moment Matching Network (github)
- MMGAN — MMGAN: Manifold Matching Generative Adversarial Network for Generating Images
- MoCoGAN — MoCoGAN: Decomposing Motion and Content for Video Generation (github)
- MPM-GAN — Message Passing Multi-Agent GANs
- MS-GAN — Temporal Coherency based Criteria for Predicting Video Frames using Deep Multi-stage Generative Adversarial Networks
- MuseGAN — MuseGAN: Symbolic-domain Music Generation and Accompaniment with Multi-track Sequential Generative Adversarial Networks
- MV-BiGAN — Multi-view Generative Adversarial Networks
- OptionGAN — OptionGAN: Learning Joint Reward-Policy Options using Generative Adversarial Inverse Reinforcement Learning
- ORGAN — Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models
- ORGAN — 3D Reconstruction of Incomplete Archaeological Objects Using a Generative Adversary Network
- PacGAN — PacGAN: The power of two samples in generative adversarial networks
- PAN — Perceptual Adversarial Networks for Image-to-Image Transformation
- PassGAN — PassGAN: A Deep Learning Approach for Password Guessing
- Perceptual GAN — Perceptual Generative Adversarial Networks for Small Object Detection
- PGAN — Probabilistic Generative Adversarial Networks
- Pip-GAN — Pipeline Generative Adversarial Networks for Facial Images Generation with Multiple Attributes
- pix2pix — Image-to-Image Translation with Conditional Adversarial Networks (github)
- PixelGAN — PixelGAN Autoencoders
- PN-GAN — Pose-Normalized Image Generation for Person Re-identification
- Pose-GAN — The Pose Knows: Video Forecasting by Generating Pose Futures
- PPGN — Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space
- PrGAN — 3D Shape Induction from 2D Views of Multiple Objects
- PSGAN — Learning Texture Manifolds with the Periodic Spatial GAN
- PS^2-GAN — High-Quality Facial Photo-Sketch Synthesis Using Multi-Adversarial Networks
- RadialGAN — RadialGAN: Leveraging multiple datasets to improve target-specific predictive models using Generative Adversarial Networks
- RAN — RAN4IQA: Restorative Adversarial Nets for No-Reference Image Quality Assessment (github)
- RankGAN — Adversarial Ranking for Language Generation
- RCGAN — Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs
- RefineGAN — Compressed Sensing MRI Reconstruction with Cyclic Loss in Generative Adversarial Networks
- RenderGAN — RenderGAN: Generating Realistic Labeled Data
- ResGAN — Generative Adversarial Network based on Resnet for Conditional Image Restoration
- RNN-WGAN — Language Generation with Recurrent Generative Adversarial Networks without Pre-training (github)
- RPGAN — Stabilizing GAN Training with Multiple Random Projections (github)
- RTT-GAN — Recurrent Topic-Transition GAN for Visual Paragraph Generation
- RWGAN — Relaxed Wasserstein with Applications to GANs

(F-R)

The GAN Zoo

Every week, new GAN papers are coming out and it's hard to keep track of them all, not to mention the incredibly creative ways in which researchers are naming these GANs! So, here's a list of what started as a fun activity compiling all named GANs!

https://github.com/hindupuravinash/the-gan-zoo

- SAD-GAN — SAD-GAN: Synthetic Autonomous Driving using Generative Adversarial Networks
- SalGAN — SalGAN: Visual Saliency Prediction with Generative Adversarial Networks (github)
- SBADA-GAN — From source to target and back: symmetric bi-directional adaptive GAN
- SCH-GAN — SCH-GAN: Semi-supervised Cross-modal Hashing by Generative Adversarial Network
- SD-GAN — Semantically Decomposing the Latent Spaces of Generative Adversarial Networks
- SEGAN — SEGAN: Speech Enhancement Generative Adversarial Network
- SeGAN — SeGAN: Segmenting and Generating the Invisible
- SegAN — SegAN: Adversarial Network with Multi-scale L1 Loss for Medical Image Segmentation
- SeqGAN — SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient (github)
- SG-GAN — Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
- SGAN — Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN — Stacked Generative Adversarial Networks (github)
- SGAN — Steganographic Generative Adversarial Networks
- SGAN — SGAN: An Alternative Training of Generative Adversarial Networks
- SimGAN — Learning from Simulated and Unsupervised Images through Adversarial Training
- SketchGAN — Adversarial Training For Sketch Retrieval
- SketchyGAN — SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis
- SL-GAN — Semi-Latent GAN: Learning to generate and modify facial images from attributes
- SN-GAN — Spectral Normalization for Generative Adversarial Networks (github)
- Sobolev GAN — Sobolev GAN
- Softmax GAN — Softmax GAN
- Splitting GAN — Class-Splitting Generative Adversarial Networks
- SRGAN — Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
- SRPGAN — SRPGAN: Perceptual Generative Adversarial Network for Single Image Super Resolution
- SS-GAN — Semi-supervised Conditional GANs
- ss-InfoGAN — Guiding InfoGAN with Semi-Supervision
- SSL-GAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- ST-CGAN — Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal
- ST-GAN — Style Transfer Generative Adversarial Networks: Learning to Play Chess Differently
- StackGAN — StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
- StarGAN — StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation (github)
- SteinGAN — Learning Deep Energy Models: Contrastive Divergence vs. Amortized MLE
- SVSGAN — SVSGAN: Singing Voice Separation via Generative Adversarial Network
- S²GAN — Generative Image Modeling using Style and Structure Adversarial Networks
- TAC-GAN — TAC-GAN — Text Conditioned Auxiliary Classifier Generative Adversarial Network (github)
- TAN — Outline Colorization through Tandem Adversarial Networks
- textGAN — Generating Text via Adversarial Training
- TextureGAN — TextureGAN: Controlling Deep Image Synthesis with Texture Patches
- TGAN — Temporal Generative Adversarial Nets
- TGAN — Tensorizing Generative Adversarial Nets
- TGAN — Tensor-Generative Adversarial Network with Two-dimensional Sparse Coding: Application to Real-time Indoor Localization
- TP-GAN — Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis
- Triple-GAN — Triple Generative Adversarial Nets
- tripletGAN — TripletGAN: Training Generative Model with Triplet Loss
- TV-GAN — TV-GAN: Generative Adversarial Network Based Thermal to Visible Face Recognition
- UGACH — Unsupervised Generative Adversarial Cross-modal Hashing
- UGAN — Enhancing Underwater Imagery using Generative Adversarial Networks
- Unrolled GAN — Unrolled Generative Adversarial Networks (github)
- VAE-GAN — Autoencoding beyond pixels using a learned similarity metric
- VariGAN — Multi-View Image Generation from a Single-View
- VAW-GAN — Voice Conversion from Unaligned Corpora using Variational Autoencoding Wasserstein Generative Adversarial Networks
- VEEGAN — VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning (github)
- VGAN — Generating Videos with Scene Dynamics (github)
- VGAN — Generative Adversarial Networks as Variational Training of Energy Based Models (github)
- VGAN — Text Generation Based on Generative Adversarial Nets with Latent Variable
- ViGAN — Image Generation and Editing with Variational Info Generative Adversarial Networks
- VIGAN — VIGAN: Missing View Imputation with Generative Adversarial Networks
- VoiceGAN — Voice Impersonation using Generative Adversarial Networks
- VRAL — Variance Regularizing Adversarial Learning
- WaterGAN — WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular Underwater Images
- WaveGAN — Synthesizing Audio with Generative Adversarial Networks
- weGAN — Generative Adversarial Nets for Multiple Text Corpora
- **WGAN — Wasserstein GAN (github)**
- **WGAN-GP — Improved Training of Wasserstein GANs (github)**
- WS-GAN — Weakly Supervised Generative Adversarial Networks for 3D Reconstruction
- XGAN — XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings
- ZipNet-GAN — ZipNet-GAN: Inferring Fine-grained Mobile Traffic Patterns via a Generative Adversarial Neural Network
- α-GAN — Variational Approaches for Auto-Encoding Generative Adversarial Networks (github)
- Δ-GAN — Triangle Generative Adversarial Networks

(S-Z)

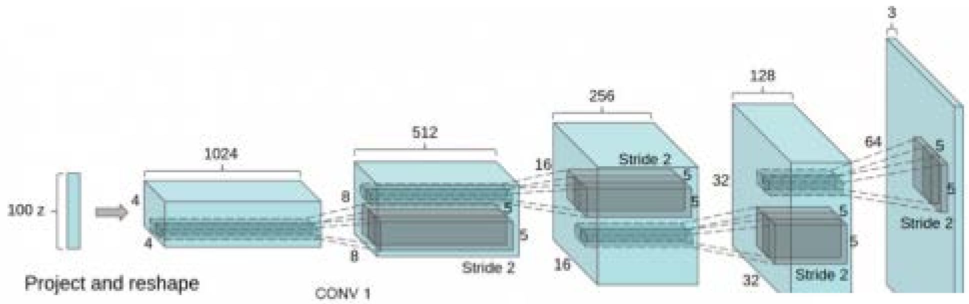
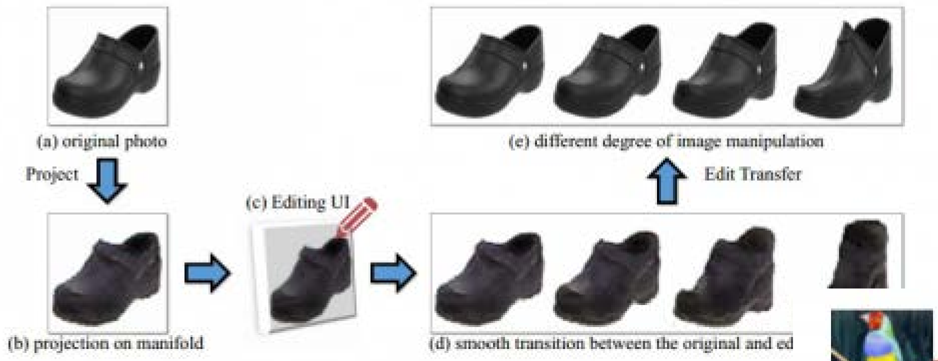
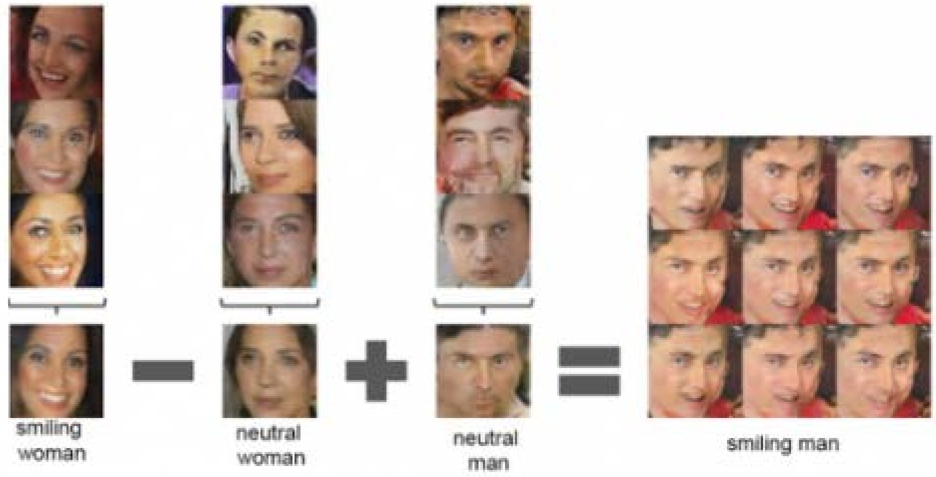


Figure 1: DCGAN generator used for LSUN scene mo
tion Z is projected to a small spatial extent convolutio
A series of four fractionally-strided convolutions (in sc
deconvolutions) then convert this high level representat
fully connected or pooling layers are used.





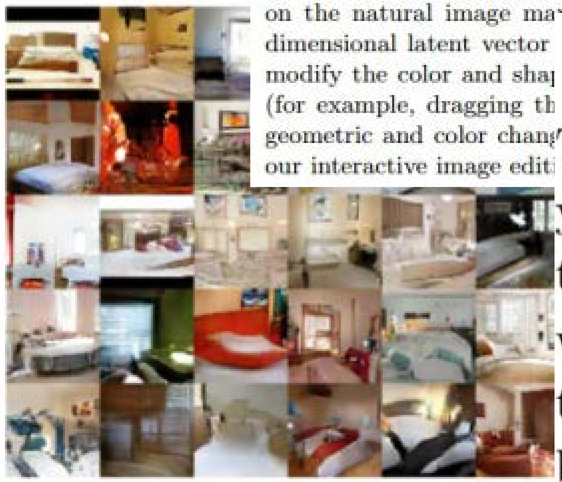
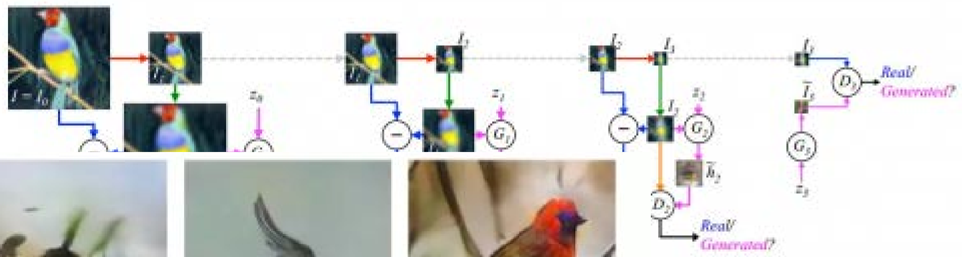


Fig. 1. We use generative adversarial networks (GAN) [1,2] to perforn
on the natural image ma
dimensional latent vector
modify the color and sha
(for example, dragging th
geometric and color chan
our interactive image edit

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face



th a 64x64 input image I from our
factor of two (red arrow) to produce
ss version l_0 of I_0; (iii) with equal
discriminative model D_0. In the real
D_0 that computes the probability of
rative network G_0 receives as input
$\tilde{h}_0 = G_0(z_0, l_0)$, which is input to
Optimizing Eqn. 2, G_0 thus learns
ss image l_0. The same procedure is
level are trained independently. At
tandard GANs G_3 & D_3.

Figure 3: Generated bedrooms after five epochs of training. There appears to be evidence of visual under-fitting via repeated noise textures across multiple samples such as the base boards of some of the beds.

Reference

1. GAN: A Beginner’s Guide to Generative Adversarial Networks .
2. Understanding Generative Adversarial Networks.
3. **通俗理解生成对抗网络GAN**
4. Generative Adversarial Nets
5. Generative Models