# Stabilizing Generative Adverserial Networks using Langevin dynamics

**Julius Ramakers**
Heinrich-Heine Universität Düsseldorf
40227 Düsseldorf, Germany
ramakers@hhu.de

**Markus Kollmann**
Heinrich-Heine Universität Düsseldorf
40227 Düsseldorf, Germany
kollmann@hhu.de

**Stefan Harmeling**
Heinrich-Heine Universität Düsseldorf
40227 Düsseldorf, Germany
harmeling@hhu.de

## Abstract

We study the problem in training Generative Adverserial Networks (GANs) both from a theoretical as well as experimental point of view. Using recent developments in mathematical formulations of generative models in terms of stochastic processes, we introduce changes to the usual optimization algorithms that lead to a better sampling performance as well as overall training success of GAN systems. We can especially avoid the problem of mode collapse. Our study is exercised on both artifical multimodial data as well as benchmark data and furthermore in combination with semi-supervised learning.

## 1 Introduction

Generative Adverserial Networks (GANs) are realised by a minimax objective, where a generator $G$ is optimised to follow the experimental data distribution as close as possible, whereas the objective of a discriminator $D$ is to separate the distributions of real and fake data, see I.J. Goodfellow et. al., 2014, [1].

The objective reads as

$$\{\phi, \theta\} = \arg\min_G \max_D J(G, D) \tag{1}$$

$$J(G, D) = \mathbb{E}_{x\sim P(x)}[\log D(x)] + \mathbb{E}_{z\sim Q(z)}[\log(1 - D(G(z)))] \tag{2}$$

with $\theta$ and $\phi$ being the parameters of the discriminator and generator, $Q(z)$ a noise distribution, and $P(x)$ the distribution of real data $x$. Usually, both $G$ and $D$ are modelled via neural networks. If we denote by $P_G(x)$ the distribution of generated examples, the optimal discriminator $D^*$ is given by

$$D^* = \frac{P(x)}{P(x) + P_G(x)} \tag{3}$$

The proof follows directly from variational calculus. Inserting the optimal discriminator into the objective gives

$$J(G, D^*) = KL(P||P_A) + KL(P_G||P_A) - 2\log 2 \tag{4}$$

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

with $P_A := (P + P_G)/2$ and

$$0 \leq J(G, D) \leq 2 \log 2 \tag{5}$$

If both $P$ and $P_G$ lie on low dimensional and non-overlapping manifolds, then $J(G, D)$ is maximal and $\nabla_\phi J(G, D) = 0$, which implies that nothing can be learned (vanishing gradient problem). To learn something we need to generate some overlap between the distributions $P$ and $P_G$.

## Main challenges for GANs

One disadvantage of GANs is that they are unstable and hard to train. Especially the problem of mode collapse and/or no overlap between $P$ and $P_G$ makes the underlying joint data distribution hard to learn, see [2] for a detailed analysis.
Furthermore, there is no knob (except regularisation strength) to tune how far the generated examples are allowed to deviate from the training set. That is because there is no way to inject 'creative noise' at right abstraction level (e.g on the level of writing style). Also the generation process is not necessary consistent. E.g. the algorithm can start with the intention to draw a 'cat' and decides along the generation process (upsampling) to draw a 'dog', resulting in a combination of both.

## Generative Modelling and adverserial training

For the actual practical code implementation of a GAN, one usually splits up the obejective function into a loss function for the generator and a loss for the discriminator. In the original formulation (see [1]), the split is

$$L_D = \mathbb{E}_{x \sim P(x)}[\log D(x)] + \mathbb{E}_{z \sim Q(z)}[\log (1 - D(G(z)))] \tag{6}$$

$$L_G = \mathbb{E}_{z \sim Q(z)}[\log (1 - D(G(z)))] \tag{7}$$

and during training one computes the gradients $\nabla L_D$ and $\nabla L_G$ on small batches of the data using some form of stochastic gradient descent. There have already been some hand-engineered solutions to now avoiding the discriminator's gradients to vanish or to avoid that the generator collapses to sharpe modes.
One can use a different measurement metrics for the distributions in the objective function to check how good the generator learns the manifold of the data. Usually those modifications target to minimize other distances than KL-divergence to especially adress the problem of mode collapse, see [3]. The most prominent modification is perhaps the Wasserstein GAN [4], which minimizes the earth-mover distance between samples from $P_G$ and $P$ and hence is more sensitive to whether the generator and true data distribution are disjoint during training. However, all of those approaches assume that training proceeds on that objective function via stochastic gradient descent. In this paper, we deviate from that assumption by addressing the fact, that in actual training of GANs one uses batches and hence, Stochastic Gradient Langevin dynamics can be applied.

# 2 Efficient sampling using Langevin dynamics

To avoid the vanishing gradient problem we need to make the data distributions broader such that they have overlapping support. This can be achieved by adding noise to the input, with the disadvantage that we would need extremely low learning rates to eventually average out noise and generate sharp pictures.
An alternative is to augment each example from the dataset by 'smeared out' versions of the same example that mimic the effect of applying noise at different strength (coarse graining). The mode collapse problem can be avoided by using an ensemble of discriminators (ideally one for each mode) instead of a single discriminator that need to be all fooled by the generator. To realize coarse graining we first pass the examples of the dataset and the generated examples through a 'coarse-grainer' such as a blur filter. We present the course-grained samples the discriminator. The generator has an easy job to fool the discriminator as modes are smeared out, gradients are smooth and data samples and generated samples have overlapping support. This also works quite good in practice since e.g. on images GANs tend to produce much too sharpe samples.

## Stochastic gradients

The concept above can be generalized by using a blur filter, $A_\lambda(x)$ to 'coarse-grain' with some blur hyperparamter $\lambda$ the level of course graining or loss of information. The objective reads then

$$\{\phi, \theta\} = \arg\min_G \max_D J(G, D) \tag{8}$$

$$J(G, D) = \mathbb{E}_{x\sim P(x)}[\log D(A_\lambda(x))] + \mathbb{E}_{z\sim Q(z)}[\log(1 - D(A_\lambda(G(z))))] \tag{9}$$

Training of the discriminator is done by stochastic gradient Langevin kind of 'Bayesian discriminator' (M. Welling, Y. Teh, 2011, [5] that results in an ensemble discriminators seen by the generator

$$\theta_{t+1} = \theta_t + \epsilon\nabla_{\theta_t}(J_t + \log P(\theta_t)) + \sqrt{2\epsilon}\eta_t \tag{10}$$

with the learning rate $\epsilon$ and noise injection $\eta_t \sim N(\eta_t|0, I_t)$. Note that the use of a Langevin equation is suboptimal as sampling is slow and the use of 'Adaptive Thermostats for Noisy Gradient Systems' (B. Leimkuhler et. al. , 2016, [6]) e.g. would be better. However, adaptive implementations tend also to be computational intensive or complex, but in practise a covariance-controlled noise level $I_t$ works well (B. Leimkuhler et. al. , 2015, [7]):

$$I_t = \left(1 - \frac{1}{t}\right) I_{t-1} + \frac{1}{t} V(\theta_t) \tag{11}$$

with $V(\theta_t)$ beeing the covariance matrix of gradients of the log likelihood. To lower the computational costs, it is also often sufficient to employ a diagonal approximation of the covariance matrix.

# 3 Experimentals

## Multimodal distributions

With our implementation, the joint gets sampled fully and also more efficient, see Fig.1.

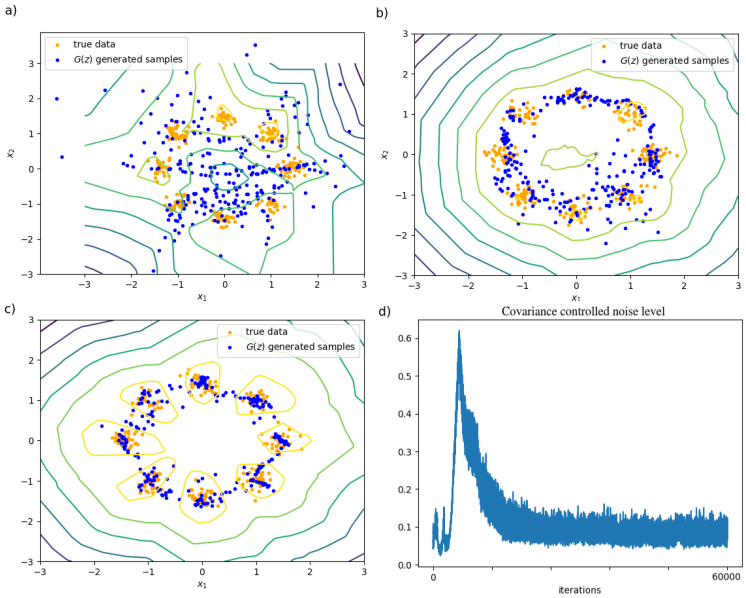


Figure 1: Samples $G(z)$ produced by our vanilla GAN from some latent variable $z$ with noisy gradient implementation. Plots show the samples along with the discriminator's lines after (a) 20000, (b) 40000 and (c) 60000 iterations. We can also see that the covariance controlled noise injection drops as desired, see bottom right (d). From the histogram one can recognize that mode collapsing does not happen in our improved setup.

Without noisy gradients, a standard GAN would produce only samples at one peak and hence collapse. For the Wasserstein GAN we can confirm that it recognizes the modes but along with covariance controlled langevin dynamics we get a good speed up compared to WGAN.

## Semi-supervised conditional GANs

We have also tested our noisy gradients implementation on a non standard GAN, namely by combining it with the semi-supervised GAN implementation by M. Mirza and S. Osindero [8]. For this implementation the blur level has to be tuned. But if one injects only a moderate blur filter and combines noisy gradients with semi-supervied GAN training, we get a good speed since the covariance controlled noise level drops over training time, see Fig.2.

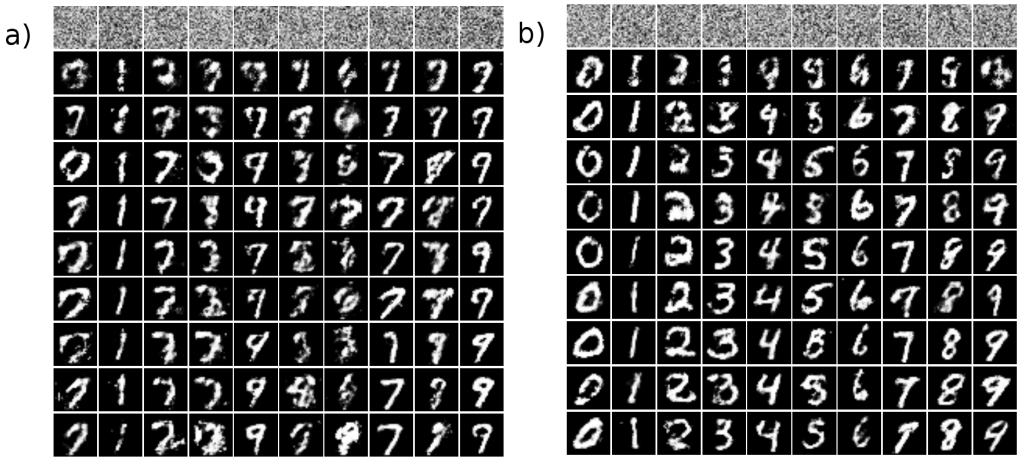


Figure 2: Semi-supervised (conditional) GAN for the MNIST dataset. We show for all MNIST numbers the generator samples after every 100000 iteartions. The images $G(z|c)$ get sampled from some latent variable $z$, but also conditional on their label $c$. On the left (a) we have a GAN with a higher blur level, on the right (b) the blur level and learning rate are in good harmony and the conditional GAN get a speed up (compared to the standard conditional version) in exploring the data distribution by using noisy covariance controlled gradients.

### References


1. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. ArXiv e-prints, June 2014.
2. M. Arjovsky and L. Bottou. Towards Principled Methods for Training Generative Ad- versarial Networks. ArXiv e-prints, January 2017.
3. I. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Scholkopf. AdaGAN: Boosting Generative Models. ArXiv e-prints, January 2017.
4. M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. ArXiv e-prints, January 2017.
5. Max Welling and Y. Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamic. Proceedings of the 28th International Conference on Machine Learning, October 2011.
6. B. Leimkuhler and X. Shang. Adaptive Thermostats for Noisy Gradient Systems. ArXiv e-prints, May 2015.
7. X. Shang, Z. Zhu, B. Leimkuhler, and A. J. Storkey. Covariance-Controlled Adaptive Langevin Thermostat for Large-Scale Bayesian Sampling. ArXiv e-prints, October 2015.
8. M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. ArXiv e-prints, November 2014.