Variational Bayesian GAN

Sung Ju Yea

Seoul National University

Department of Statistics

May 16, 2020

Introduction

- J-T. Chien and C-L. Kuo, "Variational Bayesian GAN," in *European Signal Processing Conference*, 2019.
- The goal of this paper is to resolve difficulties found in BGAN(Bayesian GAN).
- The paper tries to apply the variational inference on BGAN to seek the computational burden that BGAN encounters.

Basic Setting

- Variational Auto-Encoder(Kingma and Welling, 2014) tries to optimize a lower bound of the log likelihood.
- In VGAN(Variational Bayesian GAN), we also try to optimize the following lower bound:

Let G, D be the deterministic (neural network)function representing the generator and discriminator in a GAN setting. Then we have θ_g, θ_d as their neural network parameters respectively. Furthermore, we endow each of these parameters priors α_g, α_d respectively.

Note that we have given data $\{x_n\}$ and noise $\{z_n\}$.

Variational Inference

Then we have the following lower bound for the log likelihood:

$$\log p(x) = \log \int \int p(x, z, \theta) dz d\theta \geq E_{q(\theta|\alpha)} \left(\sum_{n=1}^{N} \left\{ E_{q_\eta(z_n|x_n)} [\log p_\theta(x_n|z_n)] - KL(q_\eta(z_n|x_n) || p(z_n)) \right\} \right) - KL(q(\theta|\alpha) || p(\theta))$$

where $\alpha = \alpha_g$ or α_d, and q is a probability density chosen for easy computation.

Also note that the lower bound is a function of α and η.

VGAN

- Hence, instead of directly dealing with the log likelihood, VGAN handles this lower bound. More precisely, the optimizing functions for G and D are given as $V_g(\eta, \alpha_g)$ and $V_d(\alpha_d)$ respectively:

$$V_g(\eta, \alpha_g) = \sum_{l=1}^{L_\theta} \left(\left[\sum_{n=1}^{N} -\left\{ E_{q_\eta(z_n|x_n)} \left[\log \frac{D(G(z_n, \theta_g^{(l)}))}{1 - D(G(z_n, \theta_g^{(l)}))} \right] + KL(q_\eta(z_n|x_n) || p(z_n)) \right\} \right] - \log q(\theta_g^{(l)}|\alpha_g) + \log p(\theta_g^{(l)}) \right)$$

VGAN

$$V_d(\alpha_d) = \sum_{l=1}^{L_\theta} \left(\sum_{n=1}^{N} \left[E_{x_n \sim p(x_n)}[\log D(x_n, \theta_d^{(l)})] + E_{z_n \sim p(z_n)}[\log(1 - D(G(z_n)), \theta_d^{(l)})] \right] + \log q(\theta_d^{(l)} | \alpha_d) - \log p(\theta_d^{(l)}) \right)$$

VGAN

There are two advantages of VGAN over BGAN that the authors propose:

- Since only α and η are involved in optimization, we do not have the posterior sampling of θ as in BGAN, which reduces computational load.
- Posterior sampling of θ requires certain amount of memory in computer. Thus, having only $\{\alpha_g, \alpha_d, \eta\}$ saves memory.

Algorithm

Algorithm 1 Learning procedure for variational GAN

Require: $\eta = \{\mu_e, \rho_e\}$, $\alpha_g = \{\mu_g, \rho_g\}$, $\alpha_d = \{\mu_d, \rho_d\}$, L_θ, k
  for number of training iterations **do**
    sample minibatch data from $p(\mathbf{x})$
    for L_θ samples **do**
      sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      calculate $\theta_g = \mu_g + \log(1 + \exp(\rho_g)) \circ \epsilon$
    end for
    update encoder η by minimizing $V_g(\eta, \alpha_g)$
    update decoder α_g by minimizing $V_g(\eta, \alpha_g)$
    for k steps **do**
      sample minibatch data from $p(\mathbf{x})$
      sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      calculate $\theta_d = \mu_d + \log(1 + \exp(\rho_d)) \circ \epsilon$
      update discriminator α_d by maximizing $V_d(\alpha_d)$
    end for
  end for

Figure: The learning algorithm for VGAN.

The authors conducted two experiments, one is using MNIST dataset and the other is using CelebA dataset. Here we briefly introduce the result of the MNIST experiment.

First, VGAN has amost always higher classification accuracy of test data versus learning epochs than BGAN.

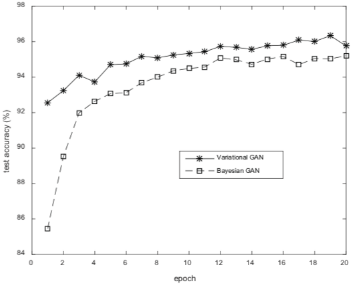

Figure: Classification accuracy of test data versus learning epochs.

Second, VGAN uses much less parameters than BGAN.

	BGAN	VGAN
generator	35748	14392
discriminator	2764	213
total	38512	14605

Figure: Comparison of number of parameters in GANs.