Bayesian GAN

Author: Saatci Y and Wilson AG

Presenter: 이종진

Seoul National University

ga0408@snu.ac.kr

may 16, 2020

- A practical Bayesian formulations for unsupervised and semi-supervised learning with GANs.
- A point mass centred on a single mode(GAN)
  / Sampling weights from posterior. (Bayesian GAN)
- Prevent mode collapse.
- Obtain good performance without any standard interventions for stabilizing GAN such as feature matching, label smoothing and mini-batch discrimination.

Unsupervised Learning

- Infer posterior over the weights:

$$p\left(\theta_g | \mathbf{z}, \theta_d\right) \propto \left(\prod_{i=1}^{n_g} D(G(z_i;\theta_g);\theta_d)\right) \times p\left(\theta_g | \alpha_g\right)$$

$$p\left(\theta_d | \mathbf{z}, \mathbf{X}, \theta_g\right) \propto \prod_{i=1}^{n_d} D(\mathbf{x}_i;\theta_d) \times \prod_{i=1}^{n_g} (1 - D(G(\mathbf{z}_i;\theta_g);\theta_d)) \times p\left(\theta_d | \alpha_d\right)$$

- Marginalizing the noise(using Monte Carlo sampling):

$$p\left(\theta_g | \theta_d\right) = \int p\left(\theta_g | \mathbf{z}, \theta_d\right) \overbrace{p\left(\mathbf{z} | \theta_d\right)}^{=p(\mathbf{z})} d\mathbf{z} \approx \frac{1}{Z_g} \sum_{j=1}^{Z_g} p(\theta_g | \mathbf{z}^{(j)}, \theta_d), \mathbf{z}^{(j)} \sim p(\mathbf{z})$$

$$p\left(\theta_d | \theta_g\right) \approx \frac{1}{Z_d} \sum_{j}^{Z_d} p(\theta_d | \mathbf{z}^{(j)}, \mathbf{X}, \theta_g), \mathbf{z}^{(j)} \sim p(\mathbf{z})$$

- We can generate sample from $p(\theta_g|\theta_d)$, $p(\theta_d|\theta_g)$ with SGHMC at every step of an epoch.(Sochastic Gradient Hamiltonian Monte Carlo).

Semisupervised Learning

- $y = 1, \dots, K$
- $y = 0$ indicates fake
- x_s, y_s mean labeled data
- Classifier C & discriminator D share same parameters θ_d
- $(m_1, \dots, m_K)$ indicates output of classifier C before softmax.
- $P(C(x;\theta_d) = k) = \frac{\exp m_k}{\sum \exp m_k}, k = 1, \dots, K$
- $P(D(x;\theta_d) = 0; \theta_d) = \frac{\sum \exp m_k}{\sum \exp m_k + 1}$
- $P(D(x;\theta_d) = 1; \theta_d) = \frac{1}{\sum \exp m_k + 1}$

Semisupervised Learning

- Infer posterior over the weights:

$$p\left(\theta_g|\mathbf{z},\theta_d\right) \propto \left(\prod_{i=1}^{n_g} D(G(\mathbf{z}_i;\theta_g)=1;\theta_d)\right) \times p\left(\theta_g|\alpha_g\right)$$

$$p\left(\theta_d|\mathbf{z},\mathbf{X},\mathbf{y}_s,\theta_g\right) \propto \prod_{i=1}^{n_d} P(D(x_i;\theta_d)=1;\theta_d) \times \prod_{i=1}^{n_g} D(G(\mathbf{z}^i;\theta_g)=0;\theta_d)$$
$$\times \left(\prod_{i=1}^{N_s} P(C(x_{is};\theta_d)=y_{is})\right) \times p\left(\theta_d|\alpha_d\right)$$

- Marginalizing the noise(Monte Carlo sampling):

$$p\left(\theta_g|\theta_d\right) \approx \frac{1}{Z_g}\sum_{j=1}^{Z_g} p(\theta_g|\mathbf{z}^{(j)},\theta_d), \mathbf{z}^{(j)} \sim p(\mathbf{z})$$

$$p\left(\theta_d|\theta_g\right) \approx \frac{1}{Z_d}\sum_{j=1}^{Z_d} p(\theta_d|\mathbf{z}^{(j)},\mathbf{X},y_s,\theta_g), \mathbf{z}^{(j)} \sim p(\mathbf{z})$$

Semisupervised Learning

- Predictive distribution for a class label y_* at a test input x_*
- Use a model average over all collected samples with respect to the posterior over θ_d

$$p\left(y_*|\mathbf{x}_*,\mathcal{D}\right) = \int p\left(y_*|\mathbf{x}_*,\theta_d\right) p\left(\theta_d|\mathcal{D}\right) d\theta_d \approx \frac{1}{T}\sum_{k=1}^{T} p\left(y_*|\mathbf{x}_*,\theta_d^{(k)}\right), \theta_d^{(k)} \sim p\left(\theta_d|\mathcal{D}\right)$$

HMC

- Metropolis Hastings algorithm may be inefficent in high-dimensional parameter space. (Random walk proposal)
- Gradient information / Introduce moment variable.
- Hamilonian Dynamics
- $U = \sum_{x \in \mathcal{D}} \log p(x|\theta) - \log p(\theta)$, $r \sim N(0, M)$

$$\pi(\theta, r) \propto \exp\left(-U(\theta) - \frac{1}{2} r^T M^{-1} r\right)$$

- Preserve hamiltonian function $U(\theta) - \frac{1}{2} r^T M^{-1} r$
- HMC simulates the Hamiltonian dynamics

$$\begin{cases} d\theta = M^{-1} r dt \\ dr = -\nabla U(\theta) dt \end{cases}$$

HMC

Algorithm: Hamiltonian Monte Carlo.

- Input: Starting position $\theta^{(1)}$ and step size ϵ

for *for* $t = 1, 2 \cdots$ *do* **do**

    $r^{(t)} \sim \mathcal{N}(0, M)$

    $(\theta_0, r_0) = \left(\theta^{(t)}, r^{(t)}\right)$

    $r_0 \leftarrow r_0 - \frac{\epsilon}{2} \nabla U(\theta_0)$

    for $i = 1$, *to* m **do**

        $\theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1}$

        $r_i \leftarrow r_{i-1} - \epsilon \nabla U(\theta_i)$

    end

    $r_m \leftarrow r_m - \frac{\epsilon}{2} \nabla U(\theta_m)$

    $(\hat{\theta}, \hat{r}) = (\theta_m, r_m)$

    Metropolis-Hastings correction:

    $u \sim$ Uniform $[0, 1]$

    $\rho = e^{H(\hat{\theta}, \hat{r}) - H\left(\theta^{(t)}, r^{(t)}\right)}$

    if $u < \min(1, \rho)$ **then**

        $\theta^{(t+1)} = \hat{\theta}$

    end

end

SGHMC with friction

- $\nabla \tilde{U}(\theta) \approx \nabla U(\theta) + \mathcal{N}(0, V(\theta))$
- HMC

$$\begin{cases} \theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1} \\ r_i \leftarrow r_{i-1} - \epsilon \nabla U(\theta_i) \end{cases}$$

- SGHMC

$$\begin{cases} \theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1} \\ r_i \leftarrow r_{i-1} - \epsilon_t M^{-1} \nabla \tilde{U}(\theta_i) - \epsilon_t C M^{-1} r_{i-1} + \mathcal{N}(0, 2(C - \hat{B})\epsilon_t) \end{cases}$$

SGHMC with friction

- SGHMC

$$\begin{cases} \theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1} \\ r_i \leftarrow r_{i-1} - \epsilon_t M^{-1} \nabla \tilde{U}(\theta_i) - \epsilon_t C M^{-1} r_{i-1} + \mathcal{N}(0, 2(C - \hat{B})\epsilon_t) \end{cases}$$

- $v = \epsilon M^{-1} r$, $\eta = \epsilon^2 M^{-1}$, $\alpha = \epsilon M^{-1} C$, $\hat{\beta} = \epsilon M^{-1} \hat{B}$

$$\begin{cases} v_i \leftarrow (1 - \alpha) v_{i-1} - \eta \nabla \tilde{U}(\theta_i) + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta) \\ \theta_i \leftarrow \theta_{i-1} + v_i \end{cases}$$

- It can be viewed as SGD with momentum $1 - \alpha$

Algorithm: Gradient discretization noise term $\hat{\beta}$ is dominated by the main friction term (this assumption constrains us to use small step sizes).
J_g: number of generator samples, J_d: number of discriminator samples
Z_g: number of noise samples
M: number of MCMC chains

- $\left\{\theta_g^{j,m}\right\}_{j=1,m=1}^{J_g,M}$, $\left\{\theta_d^{j,m}\right\}_{j=1,m=1}^{J_d,M}$ represent posteriors samples from previous iteration.

Begin

for *number of MC iterations* J_g **do**

- Sample Z_g noise samples $\left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(Z_g)}\right\}$ from noise prior $p(\mathbf{z})$. Each z $^{(i)}$ has n_g samples.
- Update sample set representing $p\left(\theta_g|\theta_d\right)$ by running SGHMC updates for M iterations:

$$\theta_g^{j,m} \leftarrow \theta_g^{j,m} + \mathbf{v}; \mathbf{v} \leftarrow (1-\alpha)\mathbf{v} + \eta \left(\sum_{i=1}^{Z_g} \sum_{k=1}^{Z_d} \frac{\partial \log p\left(\theta_g|\mathbf{z}^{(i)}, \theta_d^{k,m}\right)}{\partial \theta_g} \right) + \mathbf{n}; \mathbf{n} \sim \mathcal{N}(0, 2\alpha\eta I)$$

- Append $\theta_g^{j,m}$ to sample set.

end

Algorithm: (Continued).

for *number of MC iterations* J_d **do**

- Sample Z_d noise samples $\left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(Z_d)}\right\}$ from noise prior $p(\mathbf{z})$
- Sample minibatch of n_d data samples x.
- Update sample set representing $p\left(\theta_d|\mathbf{z}, \theta_g\right)$ by running SGHMC updates for M iterations:

$$\theta_d^{j,m} \leftarrow \theta_d^{j,m} + \mathbf{v}; \mathbf{v} \leftarrow (1-\alpha)\mathbf{v} + \eta\left(\sum_{i=1}^{Z_d}\sum_{k=1}^{Z_g}\frac{\partial \log p\left(\theta_d|\mathbf{z}^{(i)}, \mathbf{x}, \theta_g^{k,m}\right)}{\partial \theta_d}\right) + \mathbf{n}; \mathbf{n} \sim \mathcal{N}(0, 2\alpha\eta$$

- Append $\theta_d^{j,m}$ to sample set.

end

End

Sampling from posterior

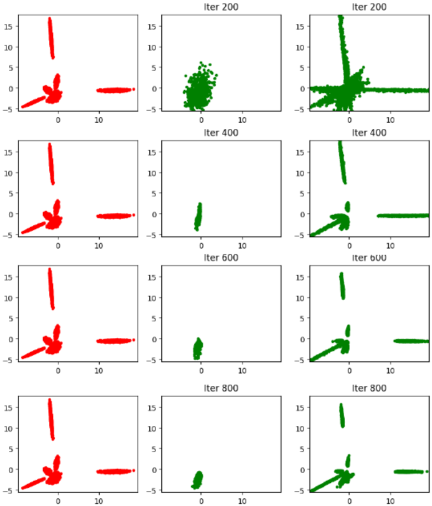
$$\theta_g^{j,m} \leftarrow \theta_g^{j,m} + \mathbf{v};\ \mathbf{v} \leftarrow (1-\alpha)\mathbf{v} + \eta \left(\sum_{i=1}^{J_g} \sum_{k=1}^{J_d} \frac{\partial \log p\left(\theta_g | \mathbf{z}^{(i)}, \theta_d^{k,m}\right)}{\partial \theta_g} \right) + \mathbf{n};\ \mathbf{n} \sim \mathcal{N}(0, 2\alpha\eta I)$$

- For generator, they implement it by SGD with momentum
- They use -log $p\left(\theta_g | \mathbf{z}^{(i)}, \theta_d^{k,m}\right) + \frac{1}{\eta}\theta n$ as objective function.
- They implement adam optimizer during first 5000 iterations.

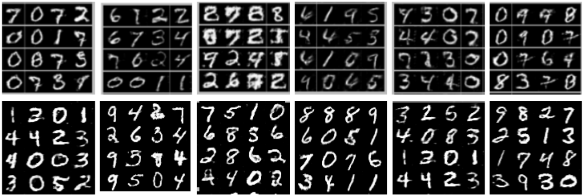
Experiments

- Comparison with standard GAN with Multimodal Synthetic data.
- $\mathbf{z} \sim \mathcal{N}(0, 10 * I_d), \quad \mathbf{A} \sim \mathcal{N}(0, I_{D\times d}),$
  $\mathbf{x} = \mathbf{Az} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.001 * I_D), d << D$
- Two-layer neural network.
- Prior $\mathcal{N}(0, I)$
- $J = 100, d = 2, D = 100$
- Plot the generated samples from Dataset disribution and each generators after applying PCA.

Experiments



Experiments



- Qualitative differences in the unsupervised data samples from Bayesian DCGAN and standard DCGAN.
- 5-layer Deconvolution GAN for generator and discriminator.
- Prior $\mathcal{N}(0, 10I)$
- $J_g = 10, J_d = 1, M = 2,\ n_d = n_g = 64$

Experiments

N_s	No. of misclassifications for MNIST. Test error rate for others.				
	Supervised	DCGAN	W-DCGAN	DCGAN-10	BayesGAN
MNIST	N=50k, $D = (28, 28)$	14	15	114	32
20	—	1823 ± 412	1687 ± 387	$\mathbf{1087 \pm 564}$	1432 ± 487
50	—	453 ± 110	490 ± 170	$\mathbf{189 \pm 103}$	332 ± 172
100	2134 ± 525	128 ± 11	156 ± 17	97 ± 8.2	$\mathbf{79 \pm 5.8}$
200	1389 ± 438	95 ± 3.2	91 ± 5.2	78 ± 2.8	$\mathbf{74 \pm 1.4}$
CIFAR-10	N=50k, $D = (32, 32, 3)$	18	19	146	68
1000	63.4 ± 2.6	58.2 ± 2.8	57.1 ± 2.4	$\mathbf{31.1 \pm 2.5}$	32.7 ± 5.2
2000	56.1 ± 2.1	47.5 ± 4.1	49.8 ± 3.1	29.2 ± 1.2	$\mathbf{26.2 \pm 4.8}$
4000	51.4 ± 2.9	40.1 ± 3.3	38.1 ± 2.9	27.4 ± 3.2	$\mathbf{23.4 \pm 3.7}$
8000	47.2 ± 2.2	29.3 ± 2.8	27.4 ± 2.5	25.5 ± 2.4	$\mathbf{21.1 \pm 2.5}$
SVHN	N=75k, $D = (32, 32, 3)$	29	31	217	81
500	53.5 ± 2.5	31.2 ± 1.8	29.4 ± 1.8	27.1 ± 2.2	$\mathbf{22.5 \pm 3.2}$
1000	37.3 ± 3.1	25.5 ± 3.3	25.1 ± 2.6	18.3 ± 1.7	$\mathbf{12.9 \pm 2.5}$
2000	26.3 ± 2.1	22.4 ± 1.8	23.3 ± 1.2	16.7 ± 1.8	$\mathbf{11.3 \pm 2.4}$
4000	20.8 ± 1.8	20.4 ± 1.2	19.4 ± 0.9	14.0 ± 1.4	$\mathbf{8.7 \pm 1.8}$
CelebA	N=100k, $D = (50, 50, 3)$	103	98	649	329
1000	53.8 ± 4.2	52.3 ± 4.2	51.2 ± 5.4	47.3 ± 3.5	$\mathbf{33.4 \pm 4.7}$
2000	36.7 ± 3.2	37.8 ± 3.4	39.6 ± 3.5	31.2 ± 1.8	$\mathbf{31.8 \pm 4.3}$
4000	34.3 ± 3.8	31.5 ± 3.2	30.1 ± 2.8	$\mathbf{29.3 \pm 1.5}$	29.4 ± 3.4
8000	31.1 ± 4.2	29.5 ± 2.8	27.6 ± 4.2	26.4 ± 1.1	$\mathbf{25.3 \pm 2.4}$