Exercises
Deep Learning
Fall 2018

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Fernando Pérez-Cruz
Web http://www.da.inf.ethz.ch/teaching/2018/DeepLearning/

Series Monday, Dec 17, 2018
(Deep Learning, Exercise series 11 - solutions)

Solution 1 (Generative Adversarial Networks):

- The failure of the generator to produce variety of samples, by collapsing too many values of z to the same sample x, is referred to as the mode collapse problem. A complete collapse is not common, but a partial collapse often happens in practice.
- Training the discriminator to optimality (i.e. the discriminator is perfectly able to distinguish real from generated samples) in the inner loop leads to a value of 0 for the objective, which in turns leads to a gradient of 0 for the generator. Thus, if the discriminator is perfect, there is no learning signal for the generator. This problem is known as the vanishing gradient problem.
- For the GAN framework to work, we need to be able to backpropagate from the discriminator's output to the generator's parameters. If the generated samples from the generator are discrete, the backpropagation can no longer proceed.
- GANs and VAEs are generative models with (complementary) differences. Some of them are listed in the table below:

feature	VAE	GAN
well defined objective (easier training, evaluation metric)	yes	no
generation of discrete data	yes	no
discrete latent variables	no	yes
sharp images/samples	no	yes

The first three problems as well as combining GANs with VAEs are open research questions in the community.

Solution 2 (Generalizing GANs to any f-divergences):

Deriving variational formulation of f-divergences.

$$D_f(P||Q) = \int_x q(x) f(\frac{p(x)}{q(x)}) dx \tag{1}$$

$$= \int_x q(x) \sup_{t \in \mathbf{dom}_{f^*}} \{t\frac{p(x)}{q(x)} - f^*(t)\} dx \tag{2}$$

$$\geq \sup_{T \in \mathrm{T}} \left(\int_x p(x) T(x) dx - \int_x q(x) f^*(T(x)) dx \right) \tag{3}$$

$$= \sup_{T \in \mathrm{T}} (\mathbb{E}_{x \sim P}[T(x)] - \mathbb{E}_{x \sim Q}[f^*(T(x))]) \tag{4}$$

Note that for the first step, we use the bi-conjugate formulation of f, $f(u) = \sup_{t \in \mathbf{dom}_{f^*}} \{tu - f^*(t)\}$. Then the lower bound is due to two reasons: *(i)* Jensen's inequality when swapping the integration and supremum operations and *(ii)* the class of functions T may contain only a subset of all possible functions.

Rewriting the lower bound using parameterized models. Assuming Q is parameterized by Θ and T is parameterized by ω, we can rewrite the lower bound as:

$$F(\Theta, \omega) = \mathbb{E}_{x \sim P}[T_\omega(x)] - \mathbb{E}_{x \sim Q_\Theta}[f^*(T_\omega(x))].$$

Learning the generative model. A generative model Q_Θ can be learned by finding a saddle-point of $F(\Theta, \omega)$ function, where we minimize with respect to Θ and maximize with respect to ω.

Reformulating with respect to the domain. Since $T_\omega = g_f(V_\omega(x))$, we can rewrite $F(\Theta, \omega)$ as:

$$F(\Theta, \omega) = \mathbb{E}_{x\sim P}[g_f(V_\omega(x))] - \mathbb{E}_{x\sim Q_\Theta}[f^*(g_f(V_\omega(x)))]. \tag{5}$$

Recovering the GAN objective. Recall the GAN objective:

$$F(\Theta, \omega) = \mathbb{E}_{x\sim P}[\log(D_\omega(x))] + \mathbb{E}_{x\sim Q_\Theta}[\log(1 - D_\omega(x)))],$$

which can be seen as a special instance of (5) by identifying each term in the expectations. In particular, choosing the last nonlinearity in the discriminator as sigmoid $D_\omega(x) = \frac{1}{1+\exp^{-V_\omega(x)}}$, corresponds to output activation function of $g_f(v) = -\log(1 + \exp^{-v})$.

If you are interested in the details of looking at GANs as a special case of a more general framework that includes all f-divergences, please refer to Nowozin et al. "f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization" arXiv:1406.2661 (2016).

Benefits of optimizing the JS divergence. In standard maximum likelihood estimation (MLE), we optimize the forward KL divergence, which is defined as $KL(P||Q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$. Since the difference between $p(x)$ and $q(x)$ is weighted by $p(x)$, whenever $p(x) = 0$, $q(x)$ is ignored, and the difference is only optimized for places where $p(x) > 0$. This leads to a model q that tends to spread its mass everywhere where $p(x) > 0$ and is not zero for places where $p(x) = 0$. This is one of the reasons why when optimizing forward KL, the samples tend to be blurry.

On the other hand, reverse KL is defined as $KL(Q|P) = \int_x q(x) \log \frac{q(x)}{p(x)} dx$. Now the difference is weighted by $q(x)$, so wherever $q(x) > 0$, the difference is optimized. However for places where $q(x) = 0$, there is no incentive to optimize the difference between the two. This leads to a model Q that focuses on certain modes of the true distribution (similarly to the mode collapse problem) as it forces $q(x) = 0$ even for places where $p(x) > 0$.

In practice, we are interested in a model that would both be able to cover the entire support of $p(x)$, and put $q(x) = 0$ for places where $p(x) = 0$. As the Jensen-Shannon divergence is defined as $JS(P||Q) = JS(Q||P) = \frac{1}{2} * KL(P||\frac{P+Q}{2}) + \frac{1}{2} * KL(Q||\frac{P+Q}{2})$, it is a combination of both the forward and reverse KL. Thus, it can be expected that minimizing the JS divergence would lead to a behavior that is in the middle of the two extremes.

Note: More details can be found on this clickable link.

Solution 3 (Practical: GANs in Tensorflow):

See the jupyter notebook: *mnist-gan.ipynb* or the python script: *mnist-gan.py*