Due Date: April 22nd 23:59, 2019
Instructions

- *Lisez toutes les questions et instruction avant de commencer.*
- *Pour toutes questions, montrez votre travail!*
- *Soumettez votre code et votre rapport (pdf) lectronique via la page du cours Gradescope. Si vous utilisez des notebook Jupyter, exportez les en pdf et soumettez les sur Gradescope.*
- *Vous pouvez mettre votre code dans un rpertoire Github et inclure le lien dans votre rapport!*

Problem 1

Les descriminateurs de modles gnratifs adversariels (GAN) estiment une mtrique entre deux distributions. Dans cette questions, vous allez implmenter un discriminateur qui estime le Jensen Shannon divergence (JSD) et un critic qui estime le Wasserstein distance (WD). Ensuite, en utilisant vos estimateurs, vous allez comparez les proprits du JSD et de la WD.

Nous fournissons des samplers [1] pour gnrer les diffrentes distributions que vous allez avoir besoin pour cette question. Vous pouvez utilisez n'importe quelle architecture que vous voulez, en autant que celles-ci ait assez de capacit pour estimer les distributions. Un MLP a trois couches devrait tre suffisant. Vous pouvez utiliser SGD avec un learning rate de $1e-3$ et des mini-batch de taille 512.

1. Implmentez une fonction pour estimer le Jensen Shanon Divergence. Le JSD est dfini comme $D_{JS}(p||q) = \frac{1}{2}D_{KL}(p||r) + \frac{1}{2}D_{KL}(q||r)$ o $r(x) = \dfrac{p+q}{2}$. Ainsi, la fonction d'objectif que votre rseau de neurone devrait optimiser est:

$$\arg\max_{\theta} \left\{ \log 2 + \frac{1}{2}\mathbf{E}_{x\sim p}[\log(D_\theta(x))] + \frac{1}{2}\mathbf{E}_{y\sim q}\log(1 - D_\theta(y))] \right\}$$

2. Implmentez une fonction pour estimer le Wasserstein Distance. La forme duel du WD est dfini comme: $W(p,q) = \sup_{||T||_L \leq 1} \mathbf{E}_{x\sim p}[T(x)] - \mathbf{E}_{y\sim q}[T(y)]$. Dans le but de contraindre vote fonction tre 1-lipschitz, nous recommandons d'utiliser la pnalit du gradient. Ainsi, l'objectif que votre rseau devrai optimiser est

$$\arg\max_{\theta} \mathbf{E}_{x\sim p}[T_\theta(x)] - \mathbf{E}_{y\sim q}[T_\theta(y)] - \lambda \mathbf{E}_{z\sim r}||\nabla_z T_\theta(z) - 1||_2.$$

   r est la distributions sur $z = ax + (1-a)y$, o $x \sim p$, $y \sim q$ et $a \sim U[0,1]$. Nous recommandons $\lambda \geq 10$.

3. Soit $Z \sim U[0,1]$ une variable alatoire suivant une distributions uniforme. Soit p une distribution dfini comme $(0, Z)$ et q_θ la distributions dfini comme (θ, Z), o θ est un paramtre. Tracez une estimation du JSD et de la WD pour $\theta \in [-1, 1]$ avec des intervales de 0.1 (c.--d. 21 points). L'axe des x devrait tre la valeur de θ et l'axe des y devrait tre votre estimation.

[1] Voir le rpertoire du devoir `IFT6135H19_assignment/assignment3/samplers.py`

4. Soit f_0 la densit d'une distribution gaussienne standard une dimension, et f_1 une fonction de densit alatoire donne par la `distribution4`. Entrainez un discriminateur D_θ en maximisant la fonction de valeur

$$\mathbb{E}_{x\sim f_1}[\log D_\theta(x)] + \mathbb{E}_{y\sim f_0}[\log(1 - D_\theta(y))]$$

Estimez la densit f_1 en utilisant la procdure de la question 5 de la partie thorique. Tracez la sortie du discriminateur et l'estimation de la densit (en utilisant le script `density_estimation.py`).

Problem 2

Les Variational Autoencoders (VAEs) sont des modles gnratifs probabilistes. Ceci veut dire qu'ils sont uttilis pour estimer $p(\boldsymbol{x})$.

Entrainez une VAE sur le dataset *Binarised MNIST*, en utilisant le ELBO ngatif comme vue en classe. La distribution priori est $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$. Chaque pixel dans le dataset est binaire: Le pixel est soit noire, soit blanc; ceci veut dire que vous devez modler la probabilit $p(\boldsymbol{x}|\boldsymbol{z})$. En d'autre mots, le dcodeur en sortie une distribution de bernouilli (ceci veut dire que vous devez utiliser la perte d'entropie croise binaire pour la reconstruction) sur les pixels. Ceci n'est pas compatible avec le dataset standard de MNIST disponible sur `torchvision`, donc soit (a) vous modifiez le code fournie avec le devoir ou, (b) vous tlchargez le dataset ici: `https://github.com/yburda/iwae/tree/master/datasets/BinaryMNIST`, et implmentez votre propre data loader.

Entrainez un VAE (10pts) Entrainez une VAE avec variable latente 100 dimensions. Ce qui suit est une suggestion d'hyperparamtres pour entraner un VAE sur Binarized MNIST. Vous pouvez utiliser votre propre architecture:

Encoder (ϕ)

Conv2d(1, 32, kernel size=(3, 3))
ELU()
AvgPool2d(kernel size=2, stride=2)
Conv2d(32, 64, kernel size=(3, 3))
ELU()
AvgPool2d(kernel size=2, stride=2)
Conv2d(64, 256, kernel size=(5, 5))
ELU()
Linear(in features=256, out features=2 $\times$ 100)
(This final layer should output mean and log-variance)

Decoder (θ)

Linear(in features=100, out features=256)
ELU()
Conv2d(256, 64, kernel size=(5, 5), padding=(4, 4))
ELU()
UpsamplingBilinear2d(scale factor=2, mode=bilinear)
Conv2d(64, 32, kernel size=(3, 3), padding=(2, 2))
ELU()
UpsamplingBilinear2d(scale factor=2, mode=bilinear)
Conv2d(32, 16, kernel size=(3, 3), padding=(2, 2))
ELU()
Conv2d(16, 1, kernel size=(3, 3), padding=(2, 2))

Ceci n'est pas du code. Utilisez l'interface de votre framework favoris pour implmenter ceci.

Utilisez ADAM avec un taux d'apprentissage de 3×10^{-4}, et entranez pour 20 epochs. valuez le modle sur un ensemble de valudation en utilise le **ELBO**. Aucun point ne va tre donne ou enlev si vous n'utilisez pas l'architecture donne. Notez que pour cette question, vous devez:

1. Obtenir en moyenne un ELBO par instance de ≥ -96 sur l'ensemble de valudation:

$$\frac{1}{|\mathcal{D}_{\text{valid}}|} \sum_{\boldsymbol{x}_i \in \mathcal{D}_{\text{valid}}} \mathcal{L}_{\text{ELBO}}(\boldsymbol{x}_i) \geq -96$$

2. Raportez le ELBO de votre modle.

Vous tre libre de modifier les hyperparamtres plus haut (sauf la taille de la variable latente) pour vous assurer que ceci fonctionne.

valuez le log-likelihood avec un VAE (20 pts) Les modles VAE sont *valus* avec log-likelihood, approxims par importance sampling, qui a t couvert durant les lectures. La formule est reproduit ci-dessous avec plus de dtails:

$$\log p(\boldsymbol{x} = \boldsymbol{x}_i) \approx \log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(\boldsymbol{x} = \boldsymbol{x}_i | \boldsymbol{z}_i^{(k)})\ p(\boldsymbol{z} = \boldsymbol{z}_i^{(k)})}{q_\phi(\boldsymbol{z} = \boldsymbol{z}_i^{(k)} | \boldsymbol{x}_i)}; \quad \text{for all } k\text{: } \boldsymbol{z}_i^{(k)} \sim q_\phi(\boldsymbol{z} | \boldsymbol{x}_i)$$

et $\boldsymbol{x}_i \in \mathcal{D}$.

1. Avec M la taille de votre *minibatch* value, $K = 200$ le nombre d'chantillon utilis pour votre importance sampling, D la dimension de votre input ($D = 784$ dans le cas de MNIST), et $L = 100$ la taille de votre variable latente, implmentez cette procdure d'importance sampling comme une fonction de:

   - **donn:**
     - votre modle entran la premire partie de la question.
     - (M, D) un array de $\boldsymbol{x}_i$'s.
     - (M, K, L) un array de $\boldsymbol{z}_{ik}$'s.
   - **retourne:**
     - $(\log p(\boldsymbol{x}_1), \ldots, \log p(\boldsymbol{x}_M))$ estime de taille $(M,)$

   Montrez le fragment de code dans votre rapport.

   Tip: Faitez attention au dbordement numrique [2] (probabilites d'une image deviennent trs petites). Utilisez le truc du LogSumExp[3] pour rgler ce problme.

2. Rapportez votre valuation de votre modle entran sur l'ensemble de validation et de test en utilisant, (a) le ELBO, et (b) l'estimation du log-likelihood ($\frac{1}{N} \sum_{i=1}^{N} \log p(\boldsymbol{x}_i)$), o N est la taille du dataset.

Assurez-vous de fournir votre code. Des points sont accords pour l'implmentation.

[2] la gnration d'un nombre trop petit pour tre reprsent sur le dispositif dsign l'emmagasin.

[3] `http://blog.smola.org/post/987977550/log-probabilities-semirings-and-floating-point`

Problem 3

Il y a prsentement un intret grandissant pour la recherche en deep learning et vision par ordinateur pour la gnration d'images. Un dfi important pour la communaut de gnration est la question d'valuation. En information, nous voulons normalement des mtriques objectives pour comparez nos algorithmes. Or, pour les tches d'esthtiques, ce n'est pas clair si un tel mtrique existe. Ainsi, valuer la qualit d'un modle gnratif demande souvent une combinaison d'valuations qualitatives et quantitatives. Ainsi, l'valuation de modles gnratifs est un dinaub de recherche actif.

Dans cette question finale, nous enqutons les avantages et les dsavantages de deux modles gnratifs populaires en deep learning: GANs et VAEs, en utilisant des mthodes d'valuations populaires de la littrature.

Street View House Numbers (SVHN) Le dataset SVHN[4] est fait d'images venant de numro de maisons house numbers de Google Street View. Comme MNIST, le but est de classifier (0-9), mais avec un dataset plus grand avec des images en couleurs.

Le dataset est disponible ici: `http://ufldl.stanford.edu/housenumbers/train_32x32.mat` (train set), et `http://ufldl.stanford.edu/housenumbers/test_32x32.mat` (test set). Sparez le training set en train et validation set comme vous voulez [5].

Alternativement, vous pouvez tlchargez, sparer et itrateurs fournis avec ce devoir.

Generative models Pour ce devoir, vous devez implmenter les **deux** modles suivants:

1. Variational Auto-Encoders (VAE):
   Consiste d'une encodeur, $f(x)$, et d'un decoder $g_{\text{VAE}}(z)$ qui retourne la moyenne d'une fonction de probabilit gaussienne $p(x|z) = \mathcal{N}(x; g_{\text{VAE}}(z), \mathbf{I})$.
2. Generative Adversarial Networks (GANs):
   Consiste d'un gnrateur $g_{\text{GAN}}(z)$, et d'un critic $d(x)$.

Notez que les deux modles ont un $g(z)$, prenant un chantillon d'une distribution a piori (dans ce cas une distribution gaussienne, $\mathcal{N}(\mathbf{0}, \mathbf{I})$), et produit une image. Ceci est primairement ce que l'valuation qualitative utilisera.

Hyperparameters & Training Pointers Entranez les deux modles avec variables latentes de dimension 100. Les deux modles ont un gnrateur prenant un code latent et produit une image. Utilisez la mme architecture pour les deux modles gmratifs

[4] `http://ufldl.stanford.edu/housenumbers/`

[5] Vous pouvez utiliser le `extra` test set disponible si vous le voulez, mais il n'y aura pas de points donnes ou enlevs si vous le faisez.

Outre ces restrictions, vous tre libre d'exprimentez avec diffrentes architecture et utilisez ce qui fonctionne. Une faon de vrfier est de regarder aux chantillon de $g(z)$. Ce qui suit sont des truc pour entraner ces modles:

- Pour GANS, utilisez l'objectif WGAN-GP avec pnalit du gradient. Vous allez peut-tre devoir ajuster le nombre de fois que vous mettez jour les paramtres du discriminateur avant de mettre jour ceux du gnrateur.
- Pour VAEs, faites attention avec la fonction d'activation que vous choisissez pour vous assurer que l'cart type est positif (exponentiel peut fonctionner, mais n'est peut-tre pas optimale pour la stabilit).
- Plus important, il y a des mtriques que vous pouvez monitorez lors de l'entranement, comme la loss du critic pour le GAN et le KL divergence entre la distribotion priori et l'approximation de la distribution a postriori pour le VAE.

Qualitative Evaluation (25pts) Nous voulons valuer le "goodness" des modles gnratif et si l'espace latent fait du sense. Par contre, ces conceptes sont subjectifs et difficiles quantifier. Ce qui suit sont des signales que les auteurs incluent dans leur papier pour dmontrer que leur modle fait quelque chose de bien.

Pour les **deux** modles:

1. **Fournissez des chantillons visuels.** Commentez, comparez et contrastez la qualit des chantillons de chaque modles (e.g. floue, diversit).
2. **Nous voulons voir si le modle appris une reprsentation dml dans l'espace latent.** chantillonez un z alatoire de votre distribution priori. Faites une petite pertubation de votre chantillon z pour *chaque dimension* (e.g. pour une dimension i, $z'_i = z_i + \epsilon$). ϵ doit tre assez large pour voir des diffrences visuelles. Pour chaque dimension, observez si le changement resulte dans des variations visuelles (cela veut dire dans $g(z)$). Vous n'avez pas besoin de montrer toutes les dimensions, juste une couple qui resulte dans des changements intressants.
3. **Comparez entre l'interpolation dans l'espace des donnes et l'espace latent.** Choisissez deux points alatoires z_0 et z_1 dans l'espace latent chantillon de la distribution priori.
   (a) Pour $\alpha = 0, 0.1, 0.2 ... 1$ calculez $z'_\alpha = \alpha z_0 + (1 - \alpha) z_1$ et tracez les chantillons rsultants $x'_\alpha = g(z'_\alpha)$.
   (b) Utilisant les chantillons des donnes $x_0 = g(z_0)$ and $x_1 = g(z_1)$ et pour $\alpha = 0, 0.1, 0.2 ... 1$ tracez les chantillons $\hat{x}_\alpha = \alpha x_0 + (1 - \alpha) x_1$.

   Expliquez la diffrence entre les deux faons d'interpoler entre les images.

valuation quantitative (25pts) Certain des meilleures modles gnratifs n'ont pas de mtrique d'valuation tractable (ils fournissent seulement des chantillons). Plusieurs mtriques sont utilises dans la communaut de recgerche pour valuer les chantillons gnrs.

Le Frechet Inception Distance (FID) est une score souvent utilis dans la littrature pour valuer la gnration de GANs. ssentielement, elle calcule la distance l_2 entre le moment de premier et de deuxime ordre entre les reprsentation gnres et la reprsentation cible. En pratique, cette representation est une couche cache d'un rseau de neuronne pr-entran sur une tche de classification.

Plus formellement, soit p un distribution cible et q la distribution venant du generateur's/decodeur's. (μ_p, Σ_p) et (μ_q, Σ_q) sont la moyenne et la covariance de la "representations" de p et q, respectivement. Le FID est dfini comme

$$d^2((\mu_p, \Sigma_p), (\mu_q, \Sigma_q)) = ||\mu_p - \mu_q||_2^2 + \mathrm{Tr}(\Sigma_p + \Sigma_q - 2(\Sigma_p\Sigma_q)^{1/2}) \tag{1}$$

Ici, nous voulons valuer les chantillons donnes par les GANs et le VAEs en utilisant le FID.

Dans le rpertoire `assignment3` du rpertoire du devoir, trouvez le script `score_fid.py`. Notre reprsentation est extraite en utilisant un classificateur custom entran sur des images de SVHN et leurs tiquettes respectives. Les TAs vous fournissent les modles entrans (`svhn_classifier.pt`), et les fonction ncssaires pour loader les images (nos loaders load les images dans le mme format que torchvision).

1. En utilisant l'quation 1, implmentez la fonction `calculate_fid_score(.,.)`, acceptant deux itrateurs en argument: un itrateur extrait les features des chantillons, et un itrateur des features extrait venant du test set.

   La fonction devrait estimer $\mu_p, \Sigma_p, \mu_q, \Sigma_q$ et calculer $d^2((\mu_p, \Sigma_p), (\mu_q, \Sigma_q))$.

2. chantillez 1000 images de $g(z)$ venant du GAN et du VAE et mettez les dans le rpertoire e.g. `path/to/sample_directory/samples/`. Assurez vous qu'il y ait seulement le rpertoire des `samples` dans le rpertoire `path/to/sample_directory/`.

   Alors, roulez `python score_fid.py path/to/sample_directory/`. Raportez le score que ce script retourne.