**Date: 10/04/2019. Time/Temps: 12h30 - 14h30**

Instructions

- *For all long-form questions, show your work!*
- *Montrez les traces de votre démarche pour toutes les questions longues!*

## Question 1 [ENGLISH] True/False *(30pts, 2 points each/chacun)*

(a) As the capacity of neural network increases, we expect the training error to increase.

(b) In training a neural network, if training set size increases, we would expect the difference between the training and generalization error to decrease.

(c) Convolving a feature map of size $(32, 32)$ with a $3 \times 4$ kernel, a stride of 2 without zero padding yields a feature map of size $(15, 15)$.

(d) The RMSProp optimizer uses momentum to accelerate learning.

(e) The Adam optimizer adapts the learning rate by using the running average of the elementwise square of gradient to estimate its second moment.

(f) Weight decay has no impact on the training of a neural network with Batch Normalization.

(g) Maximizing the ELBO wrt to the encoder $q(\boldsymbol{z}|\boldsymbol{x})$ is equivalent to minimizing the KL divergence $D_{\mathrm{KL}}(q(\boldsymbol{z}||\boldsymbol{x})||p(\boldsymbol{z}))$.

(h) PixelCNNs are efficent to train because they can exploit parallelism across pixels despite being autoregressive generative models.

(i) Reparameterization trick is used to reduced the bias in approximating the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$.

(j) The softmax activation function is shift-invariant.

(k) For some choice of the proposal distribution $q(\boldsymbol{h})$, the following inequality holds

$$\mathbb{E}_{\boldsymbol{h}}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{h})}{q(\boldsymbol{h})}\right] < \mathbb{E}_{\boldsymbol{h}_1,\dots,\boldsymbol{h}_K}\left[\frac{1}{K}\sum_{j=1}^{K} \log \frac{p(\boldsymbol{x}, \boldsymbol{h}_j)}{q(\boldsymbol{h}_j)}\right]$$

where $\boldsymbol{h}, \boldsymbol{h}_1, ..., \boldsymbol{h}_K$ are independently and identically distributed by $q(\boldsymbol{h})$ and $K > 1$.

(l) The optimal discriminator of a vanilla GAN is the density function of the real data distribution.

(m) The Jensen Shannon divergence between two distributions is always $\log 2$ whenever they have disjoint support.

(n) The Wasserstein-GAN requires regularizing the generator network to be 1-Lipschitz.

(o) An "adversarial example" is the name given to GAN generated examples that successfully fool the GAN discriminator.

## **Question 1 [FRANÇAIS]** Vrai / Faux: *(30pts, 2 points chacun)*

(a) Lorsque la capacité d'un réseau de neurone augmente, nous nous attendons à ce que l'erreur d'entraînement augmente.

(b) Dans l'entraînement d'un réseau de neuronne, si la taille de l'ensemble d'entraînement augmente, la différence entre les erreures d'entraînement et de généralisation devrait diminuer.

(c) Effectuer une convolution sur un *feature map* de taille $(32, 32)$ avec un noyau de taille $3 \times 4$, un *stride* de 2 et sans *zero padding* produit un *feature map* de taille $(15, 15)$.

(d) L'optimiseur RMSProp utilise le momentum pour accelérer l'apprentissage.

(e) L'optimiseur Adam adapte le taux d'apprentissage en utilisant la moyenne courrante (*running average*) du gradient au carré par élément pour estimer le moment d'ordre deux.

(f) Le *weight decay* n'a pas d'impacte sur l'entraînement de réseau de neuronne avec *Batch Normalization.*

(g) Maximiser le *ELBO* par rapport à l'encodeur $q(\boldsymbol{z}|\boldsymbol{x})$ est équivalent à minimiser la divergence KL $D_{\mathrm{KL}}(q(\boldsymbol{z}||\boldsymbol{x})||p(\boldsymbol{z}))$.

(h) Il est efficace d'entraîner des *PixelCNNs*, car nous pouvons exploiter le parallélisme au travers des pixels, même si ce dernier est un modèle génératif autorégressif.

(i) La technique de reparametrization (*Reparameterization trick*) est utilisée pour réduire le biais dans l'approximation de la vraie probabilité à postériori $p(\boldsymbol{z}|\boldsymbol{x})$.

(j) La fonction d'activation Softmax est invariante au décalage.

(k) Pour un choix de distribution $q(\boldsymbol{h})$, l'inégalité suivante est valable

$$\mathbb{E}_{\boldsymbol{h}}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{h})}{q(\boldsymbol{h})}\right] < \mathbb{E}_{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K}\left[\frac{1}{K} \sum_{j=1}^{K} \log \frac{p(\boldsymbol{x}, \boldsymbol{h}_j)}{q(\boldsymbol{h}_j)}\right]$$

où $\boldsymbol{h}, \boldsymbol{h}_1, ..., \boldsymbol{h}_K$ sont indépendemment et identicallement distribuée par $q(\boldsymbol{h})$ et $K > 1$.

(l) Le discriminateur optimal du GAN originale est la fonction de densité des données de la distribution des vraies données.

(m) La divergence de Jensen Shannon entre deux distributions est toujours $\log 2$ lorsque leur support est disjoint.

(n) Le generateur du Wasserstein-GAN doit être régularisé pour être 1-Lipschitz.

(o) Un "*adversarial example*" est le nom donné aux examples générés par le générateur d'un GAN qui trompent le discriminateur d'un GAN.

## **Question 2 [ENGLISH]** Short answer questions *(20pts, 2 points each)*

(a) How does one diagnose overfitting ?

(b) Explain why the capacity of a neural network grows as the number of training iterations increases.

(c) Very briefly, explain the difference between AdaGrad and RMSprop.

(d) Applying early stopping to a linear model is equivalent to what form of regularization ?

(e) Compute the full convolution (with kernel flipping) for the following 1D matrices $[1,2,3]*[1,0,1]$ (Hint: in $f * g$, the kernel is $g$).

(f) Compute the valid convolution (with kernel flipping) for the above (i.e. for $[1,2,3]*[1,0,1]$).

(g) The standard ReLU hidden unit is given by the following

$$a = \boldsymbol{w}^T \boldsymbol{x} + b, \qquad h = ReLU(a)$$

Specify the changes to the pre-activation function $a$ when we apply Batch Normalization.

(h) The BERT transformer model was trained on two self-supervised tasks, name or describe these.

(i) Given an encoder $f : \mathcal{X} \to \mathcal{H}$ and a decoder $g : \mathcal{H} \to \mathcal{X}$, write out the loss function of a contractive autoencoder. Use $L_2$ reconstruction loss.

(j) Describe the Meta-Learning evaluation setting, i.e. describe the composition of the the meta-test set and specify how we measure the Meta-Learning model performance.

## **Question 2 [FRANÇAIS]** Questions à réponse coure. *(20pts, 2 points chaque)*

(a) Comment diagnostiquer le surapprentissage (*Overfitting*) ?

(b) Expliquez pourquoi la capacitée d'un réseau de neuronne augmente avec le nombre d'itération d'entraînement.

(c) Expliquez très brièvement la différence entre AdaGrad et RMSprop.

(d) Quel type de régularisation est équivalent à appliquer la technique d'arrêt précoce (*early stopping*) à un modèle linéaire ?

(e) Calculez la convolution complète (*full convolution*) avec retournement de noyau (*kernel flipping*) pour les matrices 1D suivante $[1,2,3]*[1,0,1]$

(f) Calculez la convolution valid (*valid convolution*) avec retournement de noyau (*kernel flipping*) pour les matrices de la sous-question précédente (c.-à-d. pour $[1,2,3]*[1,0,1]$).

(g) L'unité ReLU standard est donnée par ce qui suit

$$a = \boldsymbol{w}^T \boldsymbol{x} + b, \qquad h = ReLU(a)$$

Spécifiez les changements à la fonction de pré-activation $a$ lorsque nous appliquons *Batch Normalization*.

(h) Le model transformeur BERT est entraîné sur deux tâches auto-supervisées. Nommez ou décrivez celles-ci.

(i) Soit l'encodeur $f : \mathcal{X} \to \mathcal{H}$ et décodeur $g : \mathcal{H} \to \mathcal{X}$, écrivez la fonction de coût d'un autoencodeur contractif (*contractive autoencoder*). Utilisez la fonction de coût de reconstruction $L_2$.

(j) Décrivez le cadre d'évaluation utilisé en Meta-Learning, c.-à-d. décrivez la composition du *meta-test set* et spécifiez comment nous mesurons la performance des modèles de Mete-Learning.

## **Question 3 [ENGLISH]** Supervised neural nets *(10pts)*

Let $\boldsymbol{x}$ be an $n$-dimensional vector. Recall that the softmax function $S : \boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in (0,1)^n$ is defined as $S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$.

(a) Let $\boldsymbol{x}$ be a function of vector $\boldsymbol{u}$. Show that the gradient $\nabla_{\boldsymbol{u}} \log S(\boldsymbol{x}(\boldsymbol{u}))_i$ is equal to

$$\nabla_{\boldsymbol{u}} \boldsymbol{x}(\boldsymbol{u})_i - \mathbb{E}_j[\nabla_{\boldsymbol{u}} \boldsymbol{x}(\boldsymbol{u})_j]$$

where $j$ is a random index following a categorical distribution with probability $S(\boldsymbol{x}(\boldsymbol{u}))_j$.

(b) Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be $K-$dimensional vectors related by $\boldsymbol{y} = S(\boldsymbol{x})$. Use the fact you found in the previous question to derive the gradient of the cross-entropy loss (i.e. negative log likelihood) with respect to the input of the softmax, $\nabla_{\boldsymbol{u}} L(\boldsymbol{x}, \boldsymbol{c})$, where $\boldsymbol{c}$ is a one-hot vector corresponding to the class label (i.e. a vector of all zeros except for a 1 in the position associated with the correct class):

$$L(\boldsymbol{x}, \boldsymbol{c}) = \sum_{i=1}^{K} -\boldsymbol{c}_i \log \boldsymbol{y}_i$$

## **Question 3 [FRANÇAIS]** Réseaux de neuronnes supervisés *(10pts)*

Soit $\boldsymbol{x}$ un vecteur à $n$-dimensions. Rappelez-vous que la fonction softmax $S : \boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in (0,1)^n$ est définie comme $S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$.

(a) Soit $\boldsymbol{x}$ fonction du vecteur $\boldsymbol{u}$. Montrez que le gradient $\nabla_{\boldsymbol{u}} \log S(\boldsymbol{x}(\boldsymbol{u}))_i$ est égal à

$$\nabla_{\boldsymbol{u}} \boldsymbol{x}(\boldsymbol{u})_i - \mathbb{E}_j[\nabla_{\boldsymbol{u}} \boldsymbol{x}(\boldsymbol{u})_j]$$

où $j$ est un indexe aléatoire suivant une distribution catégorique avec probabilité $S(\boldsymbol{x}(\boldsymbol{u}))_j$.

(b) Soit $\boldsymbol{y}$ et $\boldsymbol{x}$ des vecteurs à $n-$dimensions en relation par $\boldsymbol{y} = S(\boldsymbol{x})$. Utilisez le résultat obtenu à la question précédente pour dériver le gradient de *cross-entropy loss* (c.-à.-d. le *negative log likelihood*) par rapport à l'entrée du softmax, $\nabla_{\boldsymbol{u}} L(\boldsymbol{x}, \boldsymbol{c})$, où $\boldsymbol{c}$ est un vecteur *one-hot* correspondant à l'étiquette de la classe (c.-à.-d. un vecteur de 0 partout à l'exception de la position associée à la bonne classe qui est représentée par un 1):

$$L(\boldsymbol{x}, \boldsymbol{c}) = \sum_{i=1}^{K} -\boldsymbol{c}_i \log \boldsymbol{y}_i$$

## Question 4 [ENGLISH] RNNs and gradient penalty regularization *(20pts)*

Denote by $\sigma$ the logistic sigmoid function. Consider the following RNN:

$$\begin{aligned}\boldsymbol{h}_t &= \sigma(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1}) \\ y_t &= \boldsymbol{v}^\top \boldsymbol{h}_t\end{aligned}$$

Here, each $\boldsymbol{x}_t \in \mathbb{R}^d$, $\boldsymbol{h}_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}$.

Let $z_t$ be the true target of the prediction $y_t$. Consider a regularized $L_2^2$-based loss function (per time-step). The form of the regularization we will consider is inspired from the WGAN-GP and we will use it to encourage smoothness from $\boldsymbol{x}_t$ to $y_t$. Specifically, $L = \sum_t L_t$ where $L_t = (z_t - y_t)^2 + ||\nabla_{\boldsymbol{x}_t} y_t||_2^2$.

(a) Draw the computational graph for forward propagation of this RNN, unrolled for 3 time steps (from $t = 1$ to $t = 3$). Include and label the initial hidden states for the RNN: $h_0$.

(b) Express the gradient $\nabla_{\boldsymbol{h}_t} L$ recursively in terms of $\nabla_{\boldsymbol{h}_{t+1}} L$.

(c) Derive a simplified expression for the gradient $\nabla_{\boldsymbol{x}_t} y_t$.

(d) Derive simplified expressions for $\nabla_{\boldsymbol{v}} L$, $\nabla_{\boldsymbol{W}} L$ and $\nabla_{\boldsymbol{U}} L$ in terms of known quantities.

## Question 4 [FRANÇAIS] RNNs régularisation par pénalité de gradient *(20pts)*

Dénotez par $\sigma$ la fonction logistique sigmoid. Considérez le RNN suivant:

$$\begin{aligned}\boldsymbol{h}_t &= \sigma(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1}) \\ y_t &= \boldsymbol{v}^\top \boldsymbol{h}_t\end{aligned}$$

Ici, chaque $\boldsymbol{x}_t \in \mathbb{R}^d$, $\boldsymbol{h}_t \in \mathbb{R}^m$ et $y_t \in \mathbb{R}$.

Soit $z_t$ la vraie cible de la prédiction de $y_t$. Considérez une fonction de coût $L_2^2$-*based* régularisée (par pas de temps). La forme de régularisation que nous allons considérée est inspirée par WGAN-GP. Nous allons l'utiliser pour encourager le lissage (*smoothness*) de $\boldsymbol{x}_t$ à $y_t$. Spécifiquement, $L = \sum_t L_t$ où $L_t = (z_t - y_t)^2 + ||\nabla_{\boldsymbol{x}_t} y_t||_2^2$.

(a) Déssinez le graph de calcul de la propagation avant pour ce RNN, déroulé sur 3 pas de temps (de $t = 1$ à $t = 3$). Incluez et étiquettez l'état caché initial du RNN: $h_0$.

(b) Exprimez le gradient $\nabla_{\boldsymbol{h}_t} L$ récursivement en terme de $\nabla_{\boldsymbol{h}_{t+1}} L$.

(c) Dérivez une expression simplifiée pour le gradient $\nabla_{\boldsymbol{x}_t} y_t$.

(d) Dérivez une expression simplifiée pour $\nabla_{\boldsymbol{v}} L$, $\nabla_{\boldsymbol{W}} L$ and $\nabla_{\boldsymbol{U}} L$ en terme de quantitées connues.

## Question 5 [ENGLISH] VAEs *(20pts)*

Consider a latent variable model $\boldsymbol{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_K)$ where $\boldsymbol{z} \in \mathbb{R}^K$, and $\boldsymbol{x} \sim p_\theta(\boldsymbol{x}|\boldsymbol{z})$. The encoder network of variational autoencoder, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, is used to produce an approximate (variational) posterior distribution over the latent variables $\boldsymbol{z}$ for any input datapoint $\boldsymbol{x}$.

(a) Prove that the log-likelihood of the data $\log p_\theta(\boldsymbol{x})$ can be expressed as the sum of the evidence lower bound, $\mathcal{L}[q_\phi]$, and the KL divergence between $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ and $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$.

(b) Decompose the variational gap (i.e. the KL) into the approximation gap and amortization gap, and explain what they are.

(c) How might one reduce this approximation gap ?

(d) Given $K$ i.i.d. samples drawn from $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, how would you estimate the variational gap ? This method should be more accurate if a larger number of samples $K$ is used.

## Question 5 [FRANÇAIS] VAEs *(20pts)*

Considérez un modèle à variables latentes $\boldsymbol{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_K)$ où $\boldsymbol{z} \in \mathbb{R}^K$, et $\boldsymbol{x} \sim p_\theta(\boldsymbol{x}|\boldsymbol{z})$. Le réseau encodeur de l'encodeur variationel (*variational autoencoder*), $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, est utilisé pour produire une distribution à postériori approximée (*variational*) sur les variables latentes $\boldsymbol{z}$ pour n'importe quel point de donnée en entrée $\boldsymbol{x}$.

(a) Prouvez que le *log-likelihood* des données $\log p_\theta(\boldsymbol{x})$ peut être exprimé comme une somme du *evidence lower bound*, $\mathcal{L}[q_\phi]$, et de la divergence KL entre $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ et $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$.

(b) Décomposez le gap variationel (*variational gap*) (c.-à-d. le KL) en gap d'approximation et en gap d'amortization. Expliquez ce qu'ils sont.

(c) Comme est-il possible de réduire ce gap d'approximation ?

(d) Soit $K$ échantillons tiré i.i.d. de $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, comment estimeriez-vous le gap variationel ? Cette méthod est plus précise si un grand nombre $K$ d'échantillons est utilisé.