Answer Sheet (*Feuille de réponses*) : 1011
Date : 25/04/2022. Time : 9h30 - 12h30

Name / Nom : _____

Student # / Matricule : _____

Instructions [En]

- You can detach this answer page from the rest of the exam to make answering questions easier.
- For all questions, clearly write your answer in the space provided **on the answer sheet** (reverse side of this cover page).
- For true / false and multiple choice questions :
  - clearly indicate your chosen answer, with a ✓, X or • in the appropriate box or circle.
  - Any question with more than one answer indicated will be considered wrong.

Instructions [Fr]

- Vous pouvez détacher cette page de réponses du reste de l'examen pour faciliter la réponse aux questions.
- Pour toutes les questions, écrivez clairement votre réponse dans l'espace prévu **sur la feuille de réponses** (au verso de cette page de couverture).
- Pour les questions vrai/faux et à choix multiples :
  - indiquez clairement la réponse que vous avez choisie, avec un ✓, X ou • dans la case ou le cercle approprié.
  - Toute question avec plus d'une réponse indiquée sera considérée comme fausse.

1. True (*Vrai*) / False (*Faux*) *(32pts, 2 points each/chacun)* :

(a) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(b) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(c) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(d) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(e) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(f) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(g) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(h) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(i) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(j) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(k) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(l) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(m) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(n) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(o) ☐ True (*Vrai*) ... ☐ False (*Faux*)
(p) ☐ True (*Vrai*) ... ☐ False (*Faux*)

2. Convolutional Neural Networks *(16pts, 4pts each)* :

(a) $p =$ _____, $s =$ _____.
(b) $p =$ _____, $k =$ _____.
(c) $k =$ _____, $s =$ _____.
(d) $o =$ _____.

3. RNNs and backpropagation Part I *(8pts, 4pts each)* :

(i) (a) (b) (c) (d)
(ii) (a) (b) (c) (d)

4. RNNs and backpropagation Part II *(12pts, 4pts each)* :

(i) (a) (b) (c) (d)
(ii) (a) (b) (c) (d)
(iii) (a) (b) (c) (d) (e)

5. Autoregressive PixelCNN Models *(12pts, 3pts each)* :

(i) (a) (b) (c) (d) (e)
(ii) (a) (b) (c) (d) (e)
(iii) (a) (b) (c) (d) (e)
(iv) (a) (b) (c) (d) (e)

6. Normalizing Flows ***(4pts)*** : (a) (b) (c) (d) (e) (f) (g) (h)

7. Variational Autoencoders (VAEs) *(12pts, 4pts each)* :

(i) (a) (b) (c) (d) (e)
(ii) (a) (b) (c) (d)
(iii) (a) (b) (c) (d) (e)

8. Generative Adversarial Networks (GANs) *(4pts)* : (a) (b) (c) (d) (e)

1. True (*Vrai*) / False (*Faux*) *(32pts, 2 points each/chacun)*

(a) **[En]** As the capacity of neural network increases, we expect the training error to decrease.

[Fr] *Lorsque la capacité d'un réseau de neurones augmente, nous nous attendons à ce que l'erreur d'entrainement diminue.*

(b) **[En]** For a convolutional neural network with full convolutional layers, the backpropagated gradients through the convolutional layer can also be expressed as a full convolution.

[Fr] *Dans un réseau de convolutions avec convolution "full", le gradient rétropropagé à travers une couche de convolution peut aussi s'exprimer comme une convolution "full".*

(c) **[En]** Standard bias (offset) terms have no impact on the training of a neural network with Batch Normalization.

[Fr] *Les biais dans une couche d'un réseau de neurones n'a pas d'effet lorsque nous utilisons la "Batch Normalization".*

(d) **[En]** PixelCNNs are efficient because they use masked convolution to exploit parallelism across pixels during both training and generation.

[Fr] *Les PixelCNN sont efficaces car ils utilisent la convolution masquée pour exploiter le parallélisme entre les pixels lors de l'apprentissage et de la génération.*

(e) **[En]** The VAE reparameterization trick is used to enable gradient estimation in approximating the true posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$.

[Fr] *Le "reparameterization trick" de VAE est utilisé pour permettre l'estimation du gradient pour approximer la vrai posterior* $p(\boldsymbol{z} \mid \boldsymbol{x})$.

(f) **[En]** The softmax activation function is scale-invariant.
[Fr] *La fonction d'activation softmax est indépendante d'échelle.*

(g) **[En]** A transformer is a more efficient sequence model than an LSTM model because it only has a quadratic dependency on the sequence length.

[Fr] *Un "transformer" est un modèle de séquence plus efficace qu'un LSTM parce qu'il a seulement une dépendence quadratique sur la longueur de la séquence.*

(h) **[En]** Unlike an RNN, Bert-style masked language models have the advantage that they can be trained in parallel over the sequence length.

[Fr] *Contrairement au RNN, un modèle de langage masqué style "Bert" a l'avantage de pouvoir être entraîné en parallèle sur la longueur de la séquence.*

(i) **[En]** The masked language modelling objective of the Bert transformer allows it to be used directly as a generative model of language.

[Fr] *L'objectif du modèle de langage masqué du "transformer" Bert lui permet d'être utilisé directement comme un modèle génératif de langage.*

(j) **[En]** The transformer's self-attention mechanism works by computing attention weights, given by the scaled dot product between the *query* and *value* vectors, which are then combined with the *key* vectors to produce the output.

[Fr] *Le mécanisme d'auto-attention du transformateur fonctionne en calculant les poids d'attention, donnés par le produit scalaire mis à l'échelle entre les vecteurs de "query" et de "value", qui sont ensuite combinés avec les vecteurs "key" pour produire la sortie.*

(k) **[En]** For any choice of the proposal distribution $q(\boldsymbol{h})$, the following inequality holds :

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

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

[Fr] *Pour quelconque choix de distribution de proposition $q(\boldsymbol{h})$, l'inégalité suivante tient :*

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

où $\boldsymbol{h}, \boldsymbol{h}_1, ..., \boldsymbol{h}_K$ sont distribuées indépendamment et identiquement selon $q(\boldsymbol{h})$ et $K > 1$.

(l) **[En]** The gradient of the Jensen-Shannon divergence between two distributions is always zero wherever they have disjoint support.

[Fr] *Le gradient de la divergence Jensen-Shannon entre deux distributions est toujours zéro lorsque leurs supports sont disjoints.*

(m) **[En]** The disadvantage of the self-supervised learning (SSL) stategy BYOL (Bootstrap your own latents) is that it can be difficult to find the hard negative examples required for effective training.

[Fr] *Le désavantage de la stratégie de "self-supervised learning (SSL)" BYOL (Bootstrap your own latents) est qu'il peut être difficile de trouver des exemples négatifs difficiles pour un entrainement effectif.*

(n) **[En]** The SSL method Simsiam can be seen as a streamlined version of BYOL with the only major architectural difference being that Simsiam target encoder is not defined as the exponential moving average of the online encoder parameters.

[Fr] *La méthode SSL Simsiam peut être considérée comme une version simplifiée de BYOL, la principale différence architecturale étant que les paramètres de l'encodeur cible ("target") de Simsiam ne sont pas spécifiés comme une moyenne mobile exponentielle des paramètres de l'encodeur en ligne.*

(o) **[En]** The SimCLRv1 algorithm uses a large queue of stored negative examples to decouple the minibatch size from the necessity of having a large number of negative examples for optimal performance.

[Fr] *L'algorithme SimCLRv1 utilise une grande file d'attente d'exemples négatifs stockés pour découpler la taille du mini-lot de la nécessité d'avoir un grand nombre d'exemples négatifs pour des performances optimales.*

(p) **[En]** Prototypical networks can be seen both as an instance of meta-learning and as an instance of metric learning.

[Fr] *Les "Prototypical networks" peuvent être vus comme une instance du "meta-learning" et une instance du "metric learning".*

2. Convolutional Neural Networks *(16pts, 4pts each)*

[En] Assume we are given data of size $3 \times 64 \times 64$. In what follows, provide a correct configuration of a convolutional neural network layer that satisfies the specified assumption. Answer with the window size of kernel (k), stride (s), padding (p), and dilation (d, with convention $d = 1$ for no dilation). Use square windows only (e.g. same k for both width and height).

[Fr] *Assumez que nous avons des données de taille* $3 \times 64 \times 64$. *Dans ce qui suit, donnez une configuration correct d'une couche convolutionnelle qui satisfait les hypothèses spécifiées. Répondez avec une taille de fenêtre de "kernel" (k), "stride" (s), "padding" (p), et dilatation (d, avec la convention* $d = 1$ *indiquant sans dilatation). Utilisez des fenêtres carrées seulement (par exemple, même k pour la largeur et la hauteur).*

(i) **[En]** The output shape (o) of the first layer is $(64, 32, 32)$. Assume $k = 8$ without dilation. In the table below, fill in the blanks for p and s.

[Fr] *La forme de la sortie (o) de la première couche est* $(64, 32, 32)$. *Assumez* $k = 8$ *sans dilatation. Dans le tableau ci-dessous, remplissez les blancs pour p et s.*

i	p	d	k	s	o
64	?	1	8	?	32

(ii) **[En]** The output shape (o) of the first layer is $(64, 32, 32)$. Assume $d = 7$, and $s = 2$. In the table below, fill in the blanks for p and k.

[Fr] *La forme de la sortie (o) de la première couche est* $(64, 32, 32)$. *Assumez* $d = 7$, *et* $s = 2$. *Dans le tableau ci-dessous, remplissez les blancs pour p et k.*

i	p	d	k	s	o
64	?	7	?	2	32

(iii) **[En]** The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$. Specify k and s for pooling with non-overlapping window.

[Fr] *La forme de la sortie de la deuxième couche est* $(64, 8, 8)$. *Assumez* $p = 0$ *et* $d = 1$. *Spécifier k et s pour le "pooling" avec des fenêtres non chevauchantes.*

i	p	d	k	s	o
32	0	1	?	?	8

(iv) **[En]** Assume $p = 0$, $d = 1$, $k = 8$ and $s = 4$. What is output shape ?

[Fr] *Assumez* $p = 0$, $d = 1$, $k = 8$ *et* $s = 4$. *Quelle est la forme de la sortie ?*

i	p	d	k	s	o
32	0	1	8	4	?

3. RNNs and backpropagation Part I *(8pts, 4pts each)*

[En] Consider the following RNN :

$$\boldsymbol{h}_t = \tanh\left(W\boldsymbol{x}_t + U\boldsymbol{x}_{t-1} + V\boldsymbol{h}_{t-1}\right),$$

$$\hat{y}_t = \boldsymbol{q}^\top \boldsymbol{h}_t + \boldsymbol{r}^\top \boldsymbol{h}_{t-1}.$$

Here, each $\boldsymbol{x}_t \in \mathbb{R}^d$, $\boldsymbol{h}_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}$. To train this neural network we will use the objective function :

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2$$

[Fr] Considérez le RNN suivant :

$$\boldsymbol{h}_t = \tanh\left(W\boldsymbol{x}_t + U\boldsymbol{x}_{t-1} + V\boldsymbol{h}_{t-1}\right),$$

$$\hat{y}_t = \boldsymbol{q}^\top \boldsymbol{h}_t + \boldsymbol{r}^\top \boldsymbol{h}_{t-1}.$$

Ici, chaque $\boldsymbol{x}_t \in \mathbb{R}^d$, $\boldsymbol{h}_t \in \mathbb{R}^m$ et $y_t \in \mathbb{R}$. Pour entraîner ce réseau de neurones nous utiliserons la fonction objectif :

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2$$

(i) **[En]** The gradient $\nabla_{\hat{y}_t}\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is given by :
[Fr] *Le gradient* $\nabla_{\hat{y}_t}\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ *est donné par :*

(a) $\frac{y_t - \hat{y}_t}{2}$

(b) $|\hat{y}_t - y_t|$

(c) $(y_t - \hat{y}_t)^2$

(d) $\hat{y}_t - y_t$

(ii) **[En]** The gradient $\nabla_{\boldsymbol{h}_t} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is given by :
[Fr] *Le gradient* $\nabla_{\boldsymbol{h}_t} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ *est donné par :*

(a) $(\hat{y}_{t-1} - y_{t-1})\,\boldsymbol{q} + (\hat{y}_t - y_t)\,\boldsymbol{r} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t+1} + U\boldsymbol{x}_t + V\boldsymbol{h}_t\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(b) $(\hat{y}_t - y_t)\,\boldsymbol{q} + (\hat{y}_{t+1} - y_{t+1})\,\boldsymbol{r} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t+1} + U\boldsymbol{x}_t + V\boldsymbol{h}_t\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(c) $(\hat{y}_t - y_t)\,\boldsymbol{q} + (\hat{y}_{t+1} - y_{t+1})\,\boldsymbol{r} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + U\boldsymbol{x}_{t-1} + V\boldsymbol{h}_{t-1}\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(d) $(\hat{y}_t - y_t)\,\boldsymbol{q} + (\hat{y}_{t+1} - y_{t+1})\,\boldsymbol{r} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + U\boldsymbol{x}_{t-1} + V\boldsymbol{h}_{t-1}\right)\right) \odot \nabla_{\boldsymbol{h}_{t-1}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

4. RNNs and backpropagation Part II *(12pts, 4pts each)*

[En] Consider the following bidirectional RNN : / **[Fr]** *Considérez un RNN bidirectionnel :*

$$\boldsymbol{h}_t^{(f)} = \tanh\left(W\boldsymbol{x}_t + V\boldsymbol{h}_{t-1}^{(f)}\right) \qquad \boldsymbol{h}_t^{(b)} = \tanh\left(W\boldsymbol{x}_t + U\boldsymbol{h}_{t+1}^{(b)}\right) \qquad \hat{y}_t = \boldsymbol{q}^\top \boldsymbol{h}_t^{(f)} + \boldsymbol{r}^\top \boldsymbol{h}_t^{(b)}$$

[En] Here, each $\boldsymbol{x}_t \in \mathbb{R}^d$, $\boldsymbol{h}_t^{(f)} \in \mathbb{R}^m$, $\boldsymbol{h}_t^{(b)} \in \mathbb{R}^m$ and $y_t \in \mathbb{R}$. To train this neural network we will once again use the objective function : / **[Fr]** *Ici, chaque* $\boldsymbol{x}_t \in \mathbb{R}^d$,$\boldsymbol{h}_t^{(f)} \in \mathbb{R}^m$, $\boldsymbol{h}_t^{(b)} \in \mathbb{R}^m$ *et* $y_t \in \mathbb{R}$. *Pour entraîner ce réseau de neurones, nous allons encore une fois utiliser la fonction objectif* :

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{2}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$$

(i) **[En]** The gradient $\nabla_{\boldsymbol{h}_t^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is given by : / **[Fr]** *Le gradient* $\nabla_{\boldsymbol{h}_t^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ *est donné par :*

(a) $(\hat{y}_t - y_t)\,\boldsymbol{q} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t+1} + V\boldsymbol{h}_t^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(b) $(\hat{y}_{t+1} - y_{t+1})\,\boldsymbol{q} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + V\boldsymbol{h}_{t-1}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(c) $(\hat{y}_t - y_t)\,\boldsymbol{q} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + V\boldsymbol{h}_{t-1}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(d) $(\hat{y}_{t+1} - y_{t+1})\,\boldsymbol{q} + V^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t+1} + V\boldsymbol{h}_t^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t+1}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(ii) **[En]** The gradient $\nabla_{\boldsymbol{h}_t^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is given by : / **[Fr]** *Le gradient* $\nabla_{\boldsymbol{h}_t^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ *est donné par :*

(a) $(\hat{y}_{t-1} - y_{t-1})\,\boldsymbol{r} + U^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t-1} + U\boldsymbol{h}_t^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t-1}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(b) $(\hat{y}_t - y_t)\,\boldsymbol{r} + U^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + U\boldsymbol{h}_{t+1}^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t-1}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(c) $(\hat{y}_{t-1} - y_{t-1})\,\boldsymbol{r} + U^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_t + U\boldsymbol{h}_{t+1}^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t-1}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(d) $(\hat{y}_t - y_t)\,\boldsymbol{r} + U^\top\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t-1} + U\boldsymbol{h}_t^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t-1}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]$

(iii) **[En]** The gradient $\nabla_W L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is given by : / **[Fr]** *Le gradient* $\nabla_W L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ *est donné par :*

(a) $\sum_{t'=1}^{T}\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'-1} + U\boldsymbol{h}_{t'}^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top +$
$\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'+1} + V\boldsymbol{h}_{t'}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top$

(b) $\sum_{t'=1}^{T}\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'} + V\boldsymbol{h}_{t'-1}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top$

(c) $\sum_{t'=1}^{T}\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'} + U\boldsymbol{h}_{t'+1}^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top +$
$\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'} + V\boldsymbol{h}_{t'-1}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top$

(d) $\sum_{t'=1}^{T}\left[\nabla_{\boldsymbol{h}_{t'}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top + \left[\nabla_{\boldsymbol{h}_{t'}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top$

(e) $\sum_{t'=1}^{T}\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'} + U\boldsymbol{h}_{t'-1}^{(b)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(b)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top +$
$\left[\left(1 - \tanh^2\left(W\boldsymbol{x}_{t'} + V\boldsymbol{h}_{t'+1}^{(f)}\right)\right) \odot \nabla_{\boldsymbol{h}_{t'}^{(f)}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})\right]\boldsymbol{x}_{t'}^\top$

5. Autoregressive PixelCNN Models *(12pts, 3pts each)*

[En] One way to enforce autoregressive conditioning is via masking the weight parameters. Consider a two-hidden-layer convolutional neural network without kernel flipping, with kernel size 3×3 and padding size 1 on each border (so that an input feature map of size 5×5 is convolved into a 5×5 output). Define mask of type A and mask of type B as

$$(\boldsymbol{M}^A)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j < 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases} \qquad (\boldsymbol{M}^B)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j \leq 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases}$$

where the index starts from 1. Masking is achieved by multiplying the kernel with the binary mask (elementwise). Specify the receptive field of the output pixel that corresponds to the third row and the fourth column (index 34 of Figure 1 below) in each of the following 4 cases below. For each case, specify which receptive field from Figure 2 to which it corresponds.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 1 – 5×5 convolutional feature map.

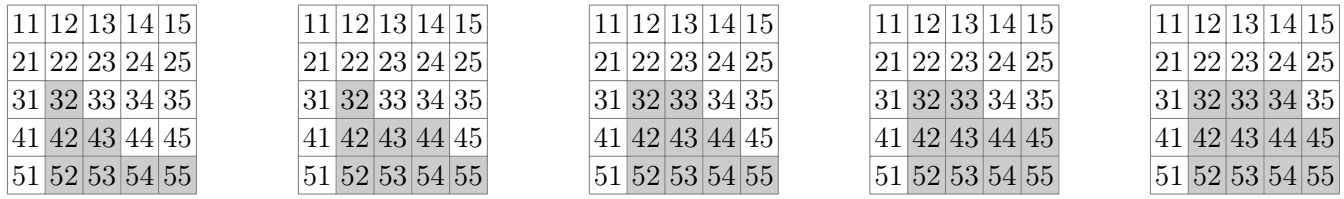

FIGURE 2 – Receptive field under different masking schemes.

[Fr] Une manière de forcer le conditionnement autorégressif est de masquer les poids du réseau. Considérez un réseau convolutif avec deux couches cachées sans reversement de noyau (sans "kernel flipping") avec noyau de taille 3×3 et "padding" de taille 1 sur chaque côté (tel qu'une entrée de taille 5×5 donne une sortie de taille 5×5). Nous définissons un masque de type A et un masque de type B comme suit :

$$(\boldsymbol{M}^A)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j < 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases} \qquad (\boldsymbol{M}^B)_{::ij} := \begin{cases} 1 & \text{if } i = 2 \text{ and } j \leq 2 \\ 1 & \text{if } i = 3 \\ 0 & \text{elsewhere} \end{cases}$$

où l'indice commence à 1. Le masquage est obtenu en multipliant le noyau avec le masque binaire (elementwise). Spécifiez le champs récepteur (receptive field) de la pixel de sortie qui correspond à la troisième rangé et quatrième colonne. (Indice 34 de la Figure 1 plus bas) dans chacun des 4 cas suivant. Pour chaque cas, spécifiez quel champs récepteur de la Figure 2 il correspond.

(i) **[En]** If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
[Fr] *Si nous utilisons* $\boldsymbol{M}^A$ *pour la première couche et* $\boldsymbol{M}^A$ *pour la seconde.*

(ii) **[En]** If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
[Fr] *Si nous utilisons* $\boldsymbol{M}^A$ *pour la première couche et* $\boldsymbol{M}^B$ *pour la seconde.*

(iii) **[En]** If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
[Fr] *Si nous utilisons* $\boldsymbol{M}^B$ *pour la première couche et* $\boldsymbol{M}^A$ *pour la seconde.*

(iv) **[En]** If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
[Fr] *Si nous utilisons* $\boldsymbol{M}^B$ *pour la première couche et* $\boldsymbol{M}^B$ *pour la seconde.*

6. Normalizing Flows *(4pts)*

[En] Normalizing flows are expressive invertible transformations of probability distributions. We assume the function $g : \mathbb{R} \to \mathbb{R}$ maps from real space to real space, and unless otherwise specified assume all parameters are real valued with no restrictions. Which of following functions is invertible on the domain $(-\infty, \infty)$?

Hint : If a function f, with range T, is *strictly monotonically increasing*[1], then there exists an inverse function f^{-1} on T. Also, if $f(z)$ and $g(z)$ are both strictly monotonically increasing on their domains then so is $g(f(z))$.

[Fr] *Les "Normalizing flows" sont des transformations inversibles expressifs d'une distribution de probabilité. Nous assumons la fonction $g : \mathbb{R} \to \mathbb{R}$, et sauf indication contraire, supposons que tous les paramètres ont une valeur réelle sans aucune restriction. Laquelle des fonctions suivantes est inversible sur le domaine $(-\infty, \infty)$?*

Indice : Lorsqu'une fonction f, avec image T, est *croissante strictement monotone*[2], alors il existe une fonction inverse f^{-1} sur T. De plus, si $f(z)$ et $g(z)$ sont tous les deux strictement monotones croissants sur leurs domaines alors il en va de même pour $g(f(z))$.

(a) **[En]** $g(z) = af\,(bz + c)$ where f is the ReLU activation function $f(x) = \max(0, x)$ and $a > 0$ and $b > 0$.

[Fr] $g(z) = af\,(bz + c)$ *où f est la fonction d'activation ReLU* $f(x) = \max(0, x)$ *et* $a > 0$ *et* $b > 0$.

(b) **[En]** $g(z) = \sigma^{-1}(\sum_{i=1}^{N} w_i \sigma(a_i z + b_i))$, $0 < w_i < 1$, where $\sum_i w_i = 1$, $a_i > 0$, and $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic sigmoid activation function and σ^{-1} is its inverse.

[Fr] $g(z) = \sigma^{-1}(\sum_{i=1}^{N} w_i \sigma(a_i z + b_i))$, $0 < w_i < 1$, *où* $\sum_i w_i = 1$, $a_i > 0$, *et* $\sigma(x) = 1/(1 + \exp(-x))$ *est la fonction logistique d'activation sigmoïde et σ^{-1} est son inverse.*

(c) **[En]** $g(z) = z + f\,(bz + c)$ where f is the ReLU activation function $f(x) = \max(0, x)$.

[Fr] $g(z) = z + f\,(bz + c)$ *où f est la fonction d'activation ReLU* $f(x) = \max(0, x)$.

(d) (a) & (b)

(e) (a) & (c)

(f) (b) & (c)

(g) All of these / *Toutes ces réponses*

(h) None of these / *Aucune de ces réponses*

1. A function may be called *strictly monotonically increasing* if and only if $x < y$ then $f(x) < f(y)$
2. Une fonction peut être appelée *croissante strictement monotone* si et seulement si $x < y$ alors $f(x) < f(y)$

7. Variational Autoencoders (VAEs) *(12pts, 4pts each)*

[En] Consider a latent variable model $\boldsymbol{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ 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 $\boldsymbol{x}$.

[Fr] *Considérez un modèle avec variable latent* $\boldsymbol{z} \sim p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ *où* $\boldsymbol{z} \in \mathbb{R}^K$, *et* $\boldsymbol{x} \sim p_\theta(\boldsymbol{x}|\boldsymbol{z})$. *Le réseau encodeur de l'autoencoder variatonnel,* $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, *est utilisé pour produire une approximation (variationnelle) de la distribution postérieure sur les variables latentes* $\boldsymbol{z}$ *étant donnée une observation* $\boldsymbol{x}$.

(i) **[En]** The log-likelihood of the data $\log p_\theta(\boldsymbol{x})$ can be expressed as the following expression :
[Fr] *La log-vraisemblance des données* $\log p_\theta(\boldsymbol{x})$ *peut être exprimée comme suit :*

(a) $\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right) - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log\left(\frac{p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}\right)$

(b) $\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log\left(\frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right) - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log\left(\frac{p(\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{x})q(\boldsymbol{z}|\boldsymbol{x})}\right)$

(c) $\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log\left(p(\boldsymbol{x})\right)$

(d) All of these / *toutes ces réponses*

(e) None of these / *aucune de ces réponses*

(ii) **[En]** The variational gap (i.e. the KL) decomposes into the approximation gap and amortization gap. One might reduce the approximation gap by :
[Fr] *L'écart variationnel (c.-à-d. la KL) se décompose en deux terms, l'écart d'approximation et l'écart d'amortissement. Il est possible de réduire l'écart d'amortissement en :*

(a) **[En]** Increasing the capacity of the encoder neural network parameterizing $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$.
[Fr] *Augmentant la capacité du réseau encodeur parameterisant* $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$.

(b) **[En]** Decreasing weight of the $\mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\|p(\boldsymbol{z}))$ term in the VAE objective function.
[Fr] *Réduisant le poids du terme* $\mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\|p(\boldsymbol{z}))$ *dans la fonction objectif du VAE.*

(c) **[En]** Including a GAN-like adversarial term to the objective function.
[Fr] *Incluant un terme adversariel style GAN à la fonction objectif.*

(d) **[En]** Use IAF in the encoder to model non-factorial distributions.
[Fr] *Utilisant IAF dans l'encodeur pour modéliser des distributions ne se factorisant pas.*

(iii) **[En]** Given K i.i.d. samples drawn from $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, how would you estimate the variational gap in practice (Assume you are computing the following over a test set of $\boldsymbol{x}$) ?
[Fr] *Étant donné* K *i.i.d. réalisations de la distribution* $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, *comment estimeriez-vous l'écart variationnel en pratique (Assumant que vous calculez ce qui suit sur un ensemble de test de* $\boldsymbol{x}$*) ?*

(a) $\log\left[\frac{1}{K}\sum_{k=1}^K \frac{p(\boldsymbol{x},\boldsymbol{z}_k)}{q_\phi(\boldsymbol{z}_k|\boldsymbol{x})}\right]$

(b) $\log\left[\frac{1}{K}\sum_{k=1}^K \frac{p(\boldsymbol{z}_k|\boldsymbol{x})}{q_\phi(\boldsymbol{z}_k|\boldsymbol{x})}\right]$

(c) $\log\left[\frac{1}{K}\sum_{k=1}^K \frac{p(\boldsymbol{x},\boldsymbol{z}_k)}{q_\phi(\boldsymbol{z}_k|\boldsymbol{x})}\right] - \frac{1}{K}\sum_{k=1}^K \left[\log p(\boldsymbol{x},\boldsymbol{z}_k) - \log q_\phi(\boldsymbol{z}_k \mid \boldsymbol{x})\right]$

(d) All of these / *Toutes ces réponses*

(e) None of these / *Aucune de ces réponses*

8. Generative Adversarial Networks (GAN) *(4pts)*

[En] Consider the dirac game in which we want to match the distribution $\mathbb{P} = \delta_0$ using the distribution $\mathbb{Q} = \delta_\theta$ and $\theta \in \mathbb{R}$. Remember that $\delta_\rho(x)$ is a dirac distribution for which $x = \rho$ with probability 1.

For this reason, we are using the Wasserstein GAN (WGAN) training framework with a critic function $C : \mathbb{R} \to \mathbb{R}$ parameterized as $C(x; \psi_0, \psi_1) = \psi_0 x + \psi_1$.

Which of the following points in the joint parameter space, (ψ_0, ψ_1, θ), are equilibria of the system which results from gradient ascent-descent optimization on the WGAN objectives ?

[Fr] Considérez le jeu dirac dans lequel nous voulons apprendre la distribution $\mathbb{P} = \delta_0$ en utilisant la distribution $\mathbb{Q} = \delta_\theta$ et $\theta \in \mathbb{R}$. Rappelons que $\delta_\rho(x)$ est une distibution dirac pour laquelle $x = \rho$ avec probabilité 1.

Pour cette raison, nous utilisons l'entrainement style Wasserstein GAN (WGAN) avec une fonction critique $C : \mathbb{R} \to \mathbb{R}$ paramétrée comme $C(x; \psi_0, \psi_1) = \psi_0 x + \psi_1$.

Lequel des points suivants dans l'espace joint de paramètres, (ψ_0, ψ_1, θ), sont des équilibres du système résultant de l'optimisation "gradient ascent-descent" sur l'objectif WGAN ?

(a) $(0, -0.5, 0)$

(b) $(0, 1, 0)$

(c) $(0, 0, 0)$

(d) All of these / *Toutes ces réponses*

(e) None of these / *Aucune de ces réponses*