

Examen IDL - IS5

Polytech Lille

January 6, 2025

PARTIE 1: QCM

Pour chaque question, veuillez entourer la réponse correspondant à votre choix. Il n'y a qu'UNE SEULE réponse correcte. Aucune explication n'est nécessaire. Aucune pénalité ne sera appliquée pour une réponse incorrecte.

(1) Qu'est-ce qui différencie le deep learning du machine learning classique ?

- Un modèle de deep learning est plus facile à expliquer qu'un modèle de machine learning classique
- Le machine learning classique nécessite une plus grande quantité de données que le deep learning
- En deep learning, on utilise des réseaux de neurones, en machine learning classique, on utilise des modèles plus simples.
- Un modèle de deep learning ne peut pas gérer des données non structurées

(2) Dans quelle situation ne faut-il PAS utiliser un modèle de deep learning ?

- La base de données contient des données non structurées (textes, images, ...)
- La base de données contient une grande quantité de données avec une grande quantité d'attributs.
- L'interprétabilité des résultats de prédiction du modèle n'est pas importante.
- L'interprétabilité des résultats de prédiction du modèle est cruciale.

(3) Parmi les réponses, quelle est l'une des principales raisons pour lesquelles les modèles de deep learning peuvent être plus performants que les modèles traditionnels dans certaines tâches ?

- Les modèles de deep learning nécessitent moins de ressources
- Ils peuvent apprendre une représentation hiérarchique de la donnée sans nécessiter d'extraire manuellement des attributs.
- Un modèle de deep learning est bien meilleur pour éviter l'overfitting

- Les modèles de deep learning ne peuvent pas être utilisés pour faire du transfer learning

(4) Pour un perceptron simple (un seul neurone artificiel), on note g la fonction d'activation, W la matrice de poids et b le vecteur de biais. Quelle est la formule liant la sortie du perceptron y à l'entrée x ?

- $y = W.g(x) + b$
- $y = g(W.x) + b$
- $y = W.x + g(b)$
- $y = g(W.x + b)$

(5) Pour un Multi-layer perceptron (MLP), quel hyperparamètre est responsable du caractère **profond** du modèle ?

- Le nombre de couches cachées
- La fonction d'activation
- Le nombre de neurones par couches
- Le nombre d'epochs

(6) Laquelle des fonctions suivantes peut être utilisée comme fonction d'activation dans la couche de sortie si l'on souhaite prédire les probabilités de n classes (p_1, p_2, \dots, p_n) de telle sorte que la somme de p sur l'ensemble des n soit égale à 1 ?

- Softmax
- Tanh
- Sigmoid
- ReLU

(7) Parmi les techniques suivantes, laquelle ne peut PAS être utilisée pour réduire l'overfitting d'un modèle de deep learning ?

- Utiliser un dropout
- Ajouter un terme de régularisation à la fonction coût
- Utiliser un early stopping
- Augmenter le nombre d'epochs

(8) Quelle affirmation est vraie parmi les suivantes ?

- Plus la taille du batch est grande, plus le processus d'optimisation est bruité.

- Plus la taille du batch est petite, moins la vitesse de calcul d'une étape de gradient est importante.
- Plus la taille du batch est petite, plus on introduit de bruit dans l'optimisation, ce qui permet de sortir plus facilement d'un minimum local.
- Plus la taille du batch est grande, moins l'optimisation est précise à chaque étape de gradient.

(9) Pourquoi le MLP n'est pas optimal pour le traitement d'images ?

- Les MLP sont conçus spécifiquement pour les données séquentielles, ce qui les rend inadaptés au traitement des images.
- Un MLP ne peut pas faire intervenir de fonctions d'activation non linéaires, qui sont essentielles pour le traitement des images.
- En utilisant un MLP directement sur une image, on perd sa structure spatiale.
- Le MLP a un champ récepteur trop localisé.

(10) Parmi les hyperparamètres suivants, lequel permet d'augmenter le champ réceptif d'un CNN ?

- Le nombre de filtres de convolution
- Le taux d'apprentissage (learning rate)
- La taille du batch de données
- La taille du filtre de convolution

(11) Quelle formule permet de calculer la taille des features maps après convolution (H_{out}, W_{out}) ? La taille du filtre (carré) est noté F , la valeur du stride est notée s .

- $H_{out} = \frac{H_{in}-F}{s} + 1$, $W_{out} = \frac{W_{in}-F}{s} + 1$
- $H_{out} = \frac{F-H_{in}}{s} + 1$, $W_{out} = \frac{F-W_{in}}{s} + 1$
- $H_{out} = \frac{s}{H_{in}-F} - 1$, $W_{out} = \frac{s}{F-W_{in}} - 1$
- $H_{out} = \frac{H_{in}}{F+s} - 1$, $W_{out} = \frac{W_{in}}{F+s} - 1$

(12) En NLP, à quoi sert le word embedding ?

- Représenter les mots sous forme de vecteurs dans un espace de grande dimension qui encodent leur fréquence dans un ensemble de données.
- Capturer le sens syntaxique et sémantique des mots en les représentant sous forme de vecteurs dans un espace de dimension plus petite.
- Classer les mots directement dans des catégories prédéfinies sans aucun traitement supplémentaire.
- S'assurer que tous les mots dont l'orthographe est similaire sont traités comme identiques dans les modèles NLP.

(13) Pour une cellule récurrente simple (dont on rappelle le schéma), quelle formule relie la sortie à l'entrée ?

- $y_t = g(W^{in}.x_t + W^h.h_t + b)$
- $y_t = W^{in}.x_t + g(W^h.h_t + b)$
- $y_t = g(W^h.x_t + W^{in}.h_t + b)$
- $y_t = g(W^h.h_t + W^{in}.x_t + b)$

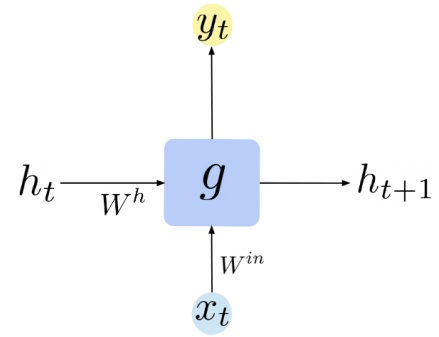


Figure 1: Cellule récurrente d'un RNN

(14) Lequel des points suivants est un avantage clé de l'utilisation des modèles Transformer dans le NLP par rapport aux réseaux neuronaux récurrents (RNN) traditionnels ?

- Les transformers utilisent un traitement séquentiel des données, ce qui les rend plus efficaces pour les longues séquences que les RNN.
- Les transformers saisissent mieux les dépendances à long terme en utilisant des mécanismes d'auto-attention, contrairement aux RNN qui s'appuient sur un traitement séquentiel.
- Les transformers utilisent des architectures plus simples, ce qui les rend moins susceptibles d'être surajoutés que les RNN.
- Les transformers utilisent un concept de portes qui sélectionnent les informations à garder ou oublier sur une séquence.

(15) Dans un autoencodeur (AE), quel est l'élément responsable de la réduction des données d'entrée en une représentation latente compressée ?

- Divergence de Kullback-Leibler (KL)
- Encoder
- Decoder
- Générateur

(16) Quelle est la différence entre un autoencodeur (AE) et un variational autoencodeur (VAE) ?

- Dans un VAE, il n'y a qu'un terme probabiliste (la divergence de Kullback-Leibler)

- Un autoencodeur apprend une représentation latente de façon déterministe, tandis qu'un autoencodeur variationnel apprend une correspondance probabiliste avec une distribution sur les variables latentes.
- Un VAE ne peut pas être entraîné en utilisant la rétropropagation.
- Un autoencodeur n'a qu'une seule couche de neurones alors qu'un variational autoencoder en a plusieurs.

(17) Dans un GAN, la fonction objectif est la suivante:

$$\min_G \max_D L(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Quel algorithme optimise les poids du discriminateur ?

- Descente de gradient
- Rétropropagation à travers le temps
- Montée de gradient
- La règle de la chaîne (chain rule)

(18) Quel est l'objectif d'un agent dans un modèle d'apprentissage par renforcement ?

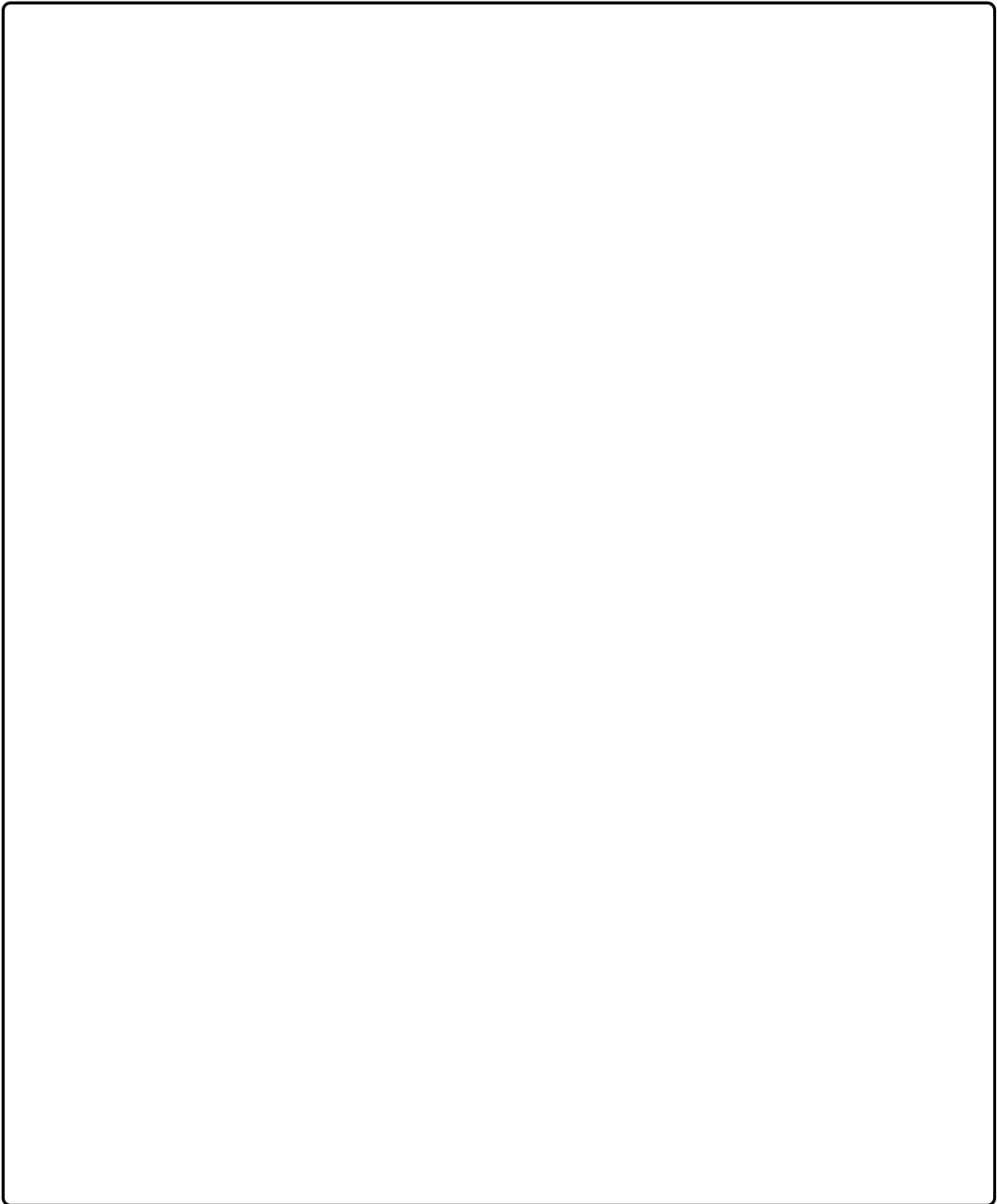
- Apprendre des actions à faire en copiant un agent exemple.
- Explorer la totalité de l'environnement, sans supervision.
- Minimiser une récompense cumulée en faisant certaines actions dans un environnement.
- Maximiser une récompense cumulée en faisant certaines actions dans un environnement.

PARTIE 2: Questions ouvertes

Répondez aux questions suivantes sur différents concepts liés au deep learning. Dessins et exemples fortement conseillés pour illustrer vos réponses. Vous pouvez utiliser une feuille supplémentaire si besoin.

(1) Qu'est-ce que la régularisation en machine learning ou deep learning, et à quoi sert-elle ? Donnez des exemples de techniques de régularisation.

(2) Qu'est ce que le mécanisme de self attention ? Expliquer ce mécanisme dans le contexte de la compréhension de texte.



(3) Qu'est-ce qu'un modèle génératif ?



PARTIE 3: Exercices

(1) Voici un schéma de principes d'un CNN. Une couche convolutionnel s'écrit de la façon suivante: $CONV2D(N_{in}, N_{out}, F)$ avec N_{in} et N_{out} respectivement le nombre canaux en entrée et en sortie. F désigne la taille du filtre utilisé.

Input: (1024, 1024, 1) (Image de résolution 1024x1024 en niveau de gris)

CONV2D(1, 16, 3): (1022, 1022, 16) (stride 1, no padding)

MAXPOOL(2): (511, 511, 16)

CONV2D(16, 32, 3): (.....,,) (stride 2, no padding)

MAXPOOL(2): (.....,,) (.....)

CONV2D(32, 64, 3): (.....,,) (stride 1, no padding)

MAXPOOL(2): (.....,,) (.....)

CONV2D(64, 128, 3): (.....,,) (stride 2, no padding)

MAXPOOL(2): (.....,,) (.....)

CONV2D(128, 256, 3): (.....,,) (stride 1, no padding)

MAXPOOL(2): (.....,,) (.....)

FLATTEN: (Flatten feature map to vector)

DENSE(9216, 512):

DROPOUT(512, 0.5):

DENSE(512, 256):

Output(1):

Consigne: Calculer les tailles de toutes les **features maps** (Compléter les pointillés). Attention à ne pas confondre avec le nombre de paramètres !

(2) Voici le schéma d'une Gated Recurrent Unit (GRU) pour l'apprentissage de séquences. Le (1 -) signifie que pour z_t en entrée, on a: $1 - z_t$ en sortie.

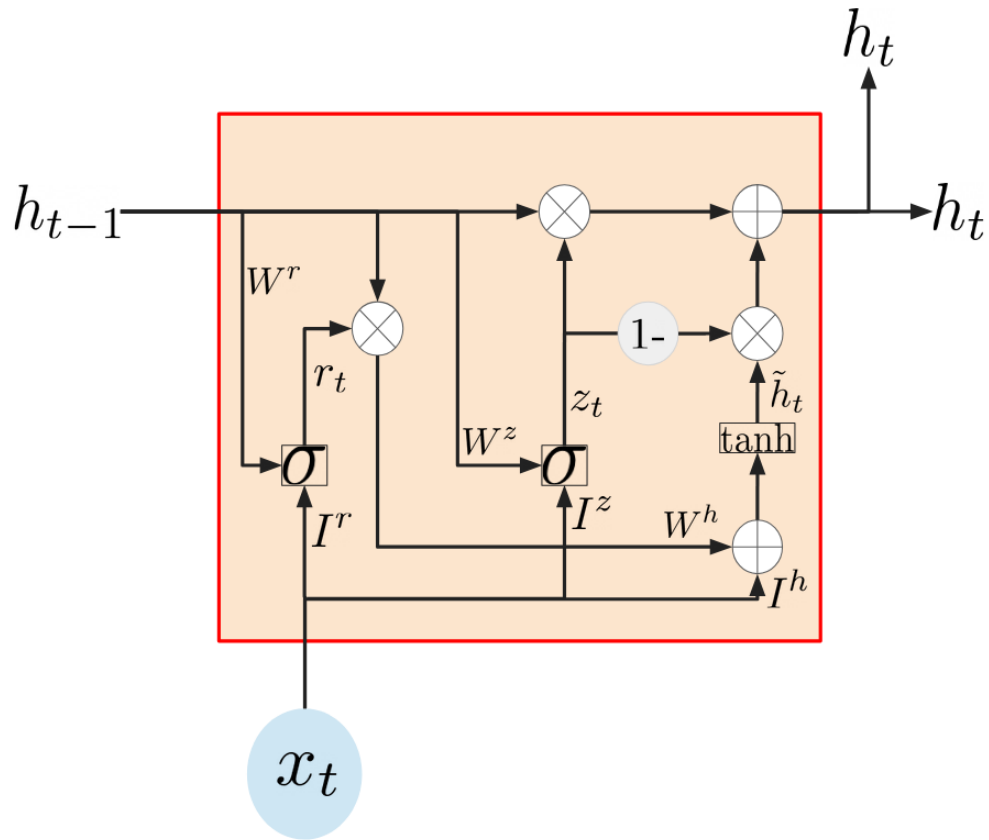


Figure 2: Schéma d'un GRU.

Compléter les formules suivantes caractérisent cette cellule. Comparer le nombre de paramètres à un RNN classique.

$$z_t = \sigma(\dots\dots\dots)$$

$$r_t =$$

$$\tilde{h}_t =$$

$$h_t =$$

(3) Voici le schéma d'un MLP simple. Une fonction coût l calcule l'erreur faite entre la prédiction \hat{y} et la valeur cible y .

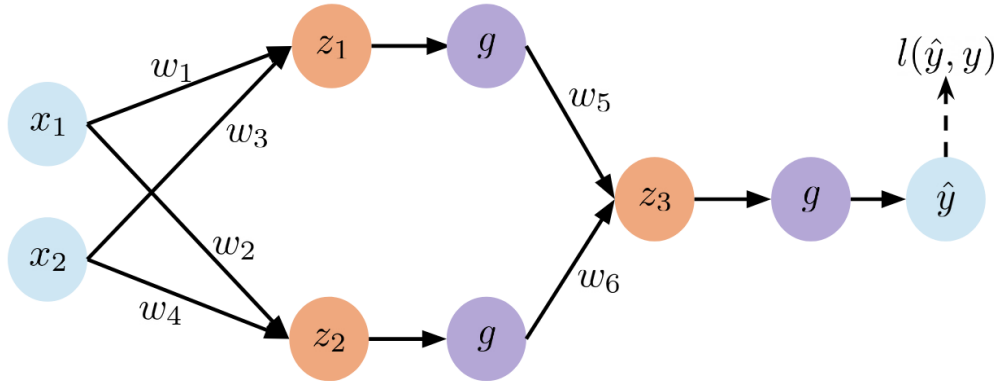


Figure 3: Schéma d'un MLP.

En particulier, **en négligeant les biais**, on a les égalités suivantes:

$$\begin{aligned} z_1 &= w_1 \cdot x_1 + w_3 \cdot x_2 \\ z_2 &= w_2 \cdot x_1 + w_4 \cdot x_2 \\ z_3 &= w_5 \cdot g(z_1) + w_6 \cdot g(z_2) \\ \hat{y} &= g(z_3) \end{aligned}$$

Rappelez brièvement en quoi consiste la backpropagation. Ecrivez les règles de la chaîne (comme dans le cours) qui permettent d'exécuter l'algorithme de backpropagation.

(Bonus) Le théorème d'approximation universel est LE résultat fondamental de la théorie du deep learning.

Théorème. Pour toute fonction continue $f : [a, b]^n \rightarrow \mathbb{R}$ et $\epsilon < 0$, il existe un MLP g avec une seule couche cachée, de sorte à ce que sa sortie $g(x)$ vérifie:

$$\sup_{x \in [a, b]^n} |f(x) - g(x)| < \epsilon \quad (1)$$

Question: Des idées pour prouver ce théorème ?