

1. Quelle est la différence entre un paramètre/poids et hyper-paramètre d'un réseau de neurones? (1 point)
 - ☐ Les paramètres/poids sont les valeurs optimisées/apprises lors d'un entraînement, alors qu'un hyper-paramètre est choisi a priori et définit l'architecture du modèle ou le type d'optimisation réalisée.
 - ☐ Un hyper-paramètre est simplement un type de poids appris pendant l'entraînement.
 - ☐ Les hyper-paramètres sont les valeurs optimisées/apprises lors d'un entraînement, alors que ce n'est pas le cas des paramètres/poids.
 - ☐ Les hyper-paramètres ne concernent pas le réseau de neurone, mais seulement le type d'optimisation réalisée.
2. Quelle fonction objectif est généralement utilisée pour une tâche de classification? (1 point)
 - ☐ Similarité cosinus.
 - ☐ Ecart absolu à la moyenne.
 - ☐ Entropie croisée.
 - ☐ Divergence Kullback-Leibler.
3. Pour quel type de tâche emploie-t-on l'erreur quadratique moyenne (Mean Squared Error) comme fonction objectif? (1 point)
 - ☐ Régression.
 - ☐ Classification.
 - ☐ Il est mieux de ne jamais utiliser l'erreur quadratique moyenne.
4. Quel ensemble de données doit être utilisé pour évaluer la performance du modèle lors d'une recherche d'hyper-paramètres? (1 point)
 - ☐ Entraînement (Train).
 - ☐ Validation.
 - ☐ Test.
5. La fonction ReLU est: (1 point)
 - ☐ Une fonction objectif.

- ☐ Une fonction de pre-processing des données.
 - ☐ Une fonction d'activation.
 - ☐ Une fonction permettant d'évaluer la performance d'un réseau de neurones.
6. Quelle est la forme de la sortie de la dernière couche d'un réseau de neurones faisant de la classification? (1 point)
- ☐ Une valeur entière correspondant à la classe.
 - ☐ Une distribution de probabilités entre les différentes classes possibles.
 - ☐ Une valeur flottante correspondant à la classe.
 - ☐ Une représentation latente non-interprétable.
7. Lors de l'exécution de l'algorithme de descente de gradient stochastique, on calcule le gradient de la fonction objectif par rapport: (1 point)
- ☐ Aux données d'entrée.
 - ☐ Aux hyper-paramètres du réseau de neurones.
 - ☐ Aux poids du réseau de neurones.
8. Quel est le bon ordre d'exécution des opérations dans l'algorithme de descente de gradient stochastique? (1 point)
- ☐ Prédiction (Forward pass) -> Échantillonnage d'un batch de données (Sampling) -> Calcul des gradients et mise à jour des poids (Backward pass) -> Calcul de l'erreur.
 - ☐ Calcul de l'erreur -> Prédiction (Forward pass) -> Calcul des gradients et mise à jour des poids (Backward pass) -> Échantillonnage d'un batch de données (Sampling).
 - ☐ Calcul de l'erreur -> Échantillonnage d'un batch de données (Sampling) -> Calcul des gradients et mise à jour des poids (Backward pass) -> Prédiction (Forward pass).
 - ☐ Échantillonnage d'un batch de données (Sampling) -> Prédiction (Forward pass) -> Calcul de l'erreur -> Calcul des gradients et mise à jour des poids (Backward pass).
 - ☐ Calcul des gradients et mise à jour des poids (Backward pass) -> Calcul de l'erreur -> Échantillonnage d'un batch de données (Sampling) -> Prédiction (Forward pass).
9. Qu'appelle-t-on biais inductifs? (1 point)
- ☐ Des termes ajoutés à une couche linéaire pour approximer une fonction affine.
 - ☐ Un ensemble d'hypothèses à propos de la fonction objectif à approximer lors de l'entraînement.
 - ☐ Des biais dans un dataset qui peuvent avoir un impact négatif sur le modèle final.
 - ☐ Des données très différentes du set d'entraînement.
10. Quel est un avantage du CNN par rapport au MLP en traitement d'images? (1 point)
- ☐ Le CNN est plus efficace en termes de nombre de paramètres à apprendre.
 - ☐ Il n'y a aucune combinaison linéaire appliquée dans un CNN.
 - ☐ Le CNN permet à des pixels très éloignés dans l'image de partager de l'information dès la première couche du réseau.
11. Les tenseurs en entrée et sortie d'une couche de convolution possèdent deux dimensions spatiales (hauteur et largeur). Soient I_x et O_x les tailles des tenseurs d'entrée et sortie d'une couche de convolution le long de l'axe x (un des deux axes, hauteur ou largeur). Notons ensuite la taille du noyau de convolution, le padding et le stride K_x , P_x et S_x le long de ce même axe x . Quelle est la bonne relation? (1 point)

- ☐ $O_x = \frac{I_x - K_x + 2P_x}{S_x}$
- ☐ $O_x = I_x - K_x + 2P_x$
- ☐ $O_x = I_x - K_x + 2P_x) - 1$
- ☐ $O_x = 1 + \frac{I_x - K_x + P_x}{S_x}$
- ☐ $O_x = 1 + (I_x - K_x + P_x)$
- ☐ $O_x = 1 + \frac{I_x - K_x + 2P_x}{S_x}$
- ☐ $O_x = I_x - 2K_x + P_x$
- ☐ $O_x = 1 + \frac{I_x - 2K_x + P_x}{S_x}$

12. Pytorch: "torch.nn.Conv2d" possède un paramètre appelé "out_channels". A quoi correspond-il? (1 point)

- ☐ La taille des noyaux de convolution.
- ☐ Le nombre de cartes de caractéristiques en entrée.
- ☐ Le nombre de noyaux de convolution.

13. A quoi correspond le champ réceptif d'un neurone en traitement d'image? (1 point)

- ☐ La surface de l'image d'entrée dont le neurone concerné reçoit de l'information.
- ☐ Le nombre de neurones de la couche suivante avec lesquels le neurone communique.
- ☐ Le nombre d'images dans le dataset pour lesquelles le neurone s'active.

14. Quel est l'intérêt de traiter les entrées d'un réseau de neurones sous forme d'un batch? (1 point)

- ☐ Un batch d'entrées prend moins de place en mémoire qu'une entrée seule.
- ☐ Pouvoir prédire les sorties associées à un ensemble d'entrées différentes en une seule passe (forward pass).
- ☐ Chaque sortie associée à une entrée du batch sera différente de la sortie obtenue si elle était traitée seule.
- ☐ En pratique, on ne peut pas prédire la sortie associée à une seule entrée à la fois avec un réseau de neurones.

15. Comment un RNN garde-t-il une mémoire du passé? (1 point)

- ☐ En mettant à jour et stockant un état caché à l'instant t qui sera repassé en entrée du modèle à l'instant t+1.
- ☐ Grâce à un encodage des instants temporels (positional encodings).
- ☐ A l'aide d'un pré-traitement et post-traitement des séquences pour inclure de l'information temporelle.

16. Quelle est une limitation du RNN classique? (1 point)

- ☐ Il est constitué de trop peu de poids et ne permet donc pas d'approximer des fonctions suffisamment complexes.
- ☐ En pratique, le RNN a des difficultés à mémoriser l'information qui provient d'un contexte lointain.
- ☐ En pratique, le RNN a des difficultés à mémoriser l'information qui provient d'un contexte proche car la mise à jour du contenu de sa mémoire est lente.

17. Qu'appelle-t-on une porte (gate) dans certaines approches comme le LSTM? (1 point)

- ☐ Un mécanisme pour sélectionner l'information à laisser passer.

- ☐ Un mécanisme pour stopper la rétropropagation des gradients temporels.
- ☐ Un mécanisme pour stocker l'information passée.
- ☐ Un mécanisme pour ajouter un signal de supervision auxiliaire à l'entraînement du LSTM.

18. Combien de portes composent le LSTM?

(1 point)

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 6

19. Pytorch: considérons les variables "optimizer" de type "torch.optim.Optimizer" et "loss" de type "torch.Tensor". "loss" est le résultat du calcul des erreurs de votre modèle par rapport à une vérité terrain (en utilisant par exemple une instance de "torch.nn.CrossEntropyLoss"). Quelle opération met à jour les paramètres de votre réseau de neurones?

(1 point)

- ☐ loss.backward()
- ☐ optimizer.step()
- ☐ loss.step()
- ☐ optimizer.backward()

20. Pytorch: Lorsqu'on implémente une classe héritant de "torch.utils.data.Dataset", quelles sont les 2 méthodes à surcharger en plus de "__init__"?

(1 point)

- ☐ __len__, __getitem__
- ☐ __len__, forward
- ☐ forward, __getitem__
- ☐ forward, __data__
- ☐ __len__, __load__
- ☐ __len__, __loaditem__
- ☐ __type__, __loaditem__