Découverte de l'Intelligence Artificielle

Examen sur table : Option L1 et CPES

Simon Collet, Thomas Deneux, George Marchment

Date : 21/12/2023

Numéro de copie anonymisée :

Exercice 1 : Questions de cours

1. Définissez chacun des termes suivants en quelques lignes chacun.

- Modèle (dans le contexte du Machine Learning)

- Apprentissage supervisé

- Apprentissage non supervisé

- Apprentissage par renforcement

2. Pour chacun des trois algorithmes suivants, précisez s'ils relèvent de l'apprentissage supervisé, non supervisé ou par renforcement et expliquez leur principe en une courte phrase (sans entrer dans aucun détail mathématique).

- KNN

- Réseau de neurones artificiels

- Q-learning

3. Qu'appelle-t-on les "biais des IA" ? Nous avons détaillé en cours deux causes distinctes de biais, dans le cadre de l'apprentissage supervisé. Expliquez ces deux causes en donnant à chaque fois un exemple dans le cas d'une tâche de reconnaissance d'image.

Exercice 2 : Interprétation code Python

Analysez le code Python ci-dessous, et répondez aux questions.

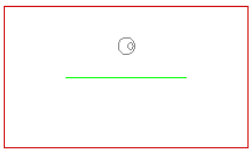
```
def fonction_mystere(liste):
    indice = 0
    val = liste[0]
    for i in range(1, len(liste)):
        if liste[i] > val:
            indice = i
            val = liste[i]
    return indice

ma_liste = [12, 45, 67, 23, 56, 89, 34]
val_mystere = fonction_mystere(ma_liste)
```

1. Quelle tâche effectue la fonction “`fonction_mystere`” ?

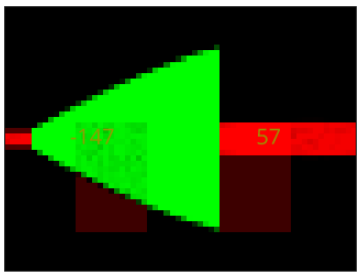
2. Que contient la variable “`val_mystere`” après exécution de ce code ?

Exercice 3 : Interprétation de l'espace d'états

Ci-dessous, nous reprenons la visualisation de l'espace des états d'entrée de l'IA comme dans le cas du dernier TP. Pour rappel, voici le plan de l'arène :

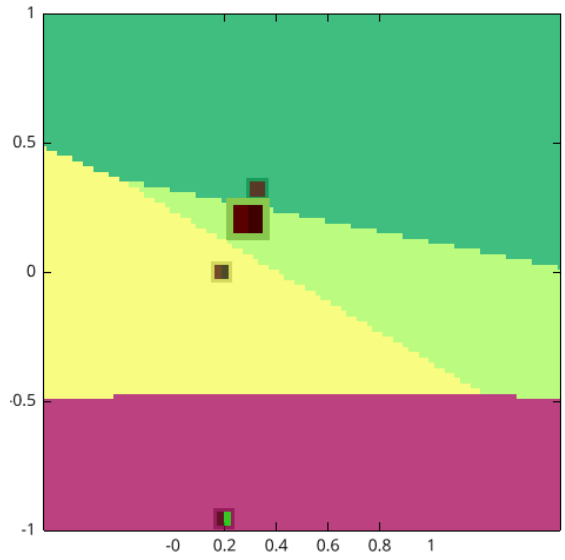


Notre robot simulé reçoit 2 entrées qui sont le calcul "canal rouge - canal vert" pour 2 zones de l'image à gauche et à droite de son champ visuel. Dans l'exemple ci-dessous, ces deux valeurs sont -147 et 57. La première valeur, négative, correspond à la présence d'un mur vert (mur intérieur). La seconde valeur, positive, correspond à un mur rouge (mur extérieur). Une valeur proche de zéro correspond à une quantité équivalente de rouge et de vert.

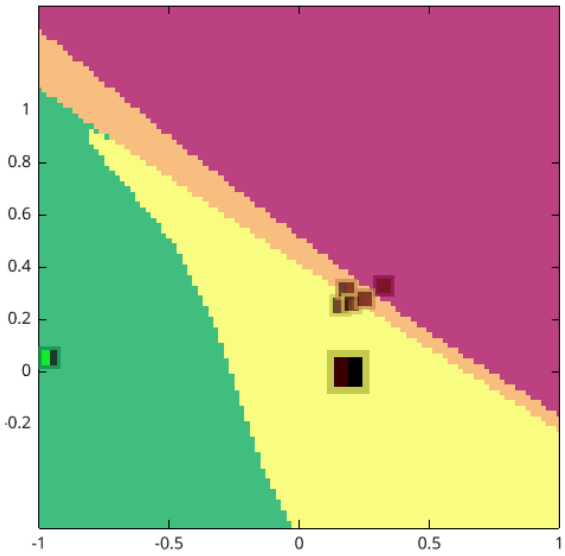


Les valeurs sont comprises entre -255 et 255 sur l'image de la caméra mais sont renormalisées entre -1 et 1 sur le graphe d'états.

Ci-dessous, nous représentons deux graphes d'états différents suite à l'entraînement de deux algorithmes distincts.



Entraînement n°1

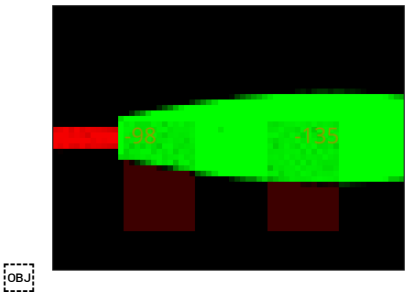


Entraînement n°2

Les couleurs des points et de l'arrière-plan correspondent aux actions disponibles au robot ci-dessous ; les points carrés sont les données d'entraînement (le point plus gros n'est pas une donnée d'entraînement, mais dans les deux cas il cache une donnée d'entraînement).

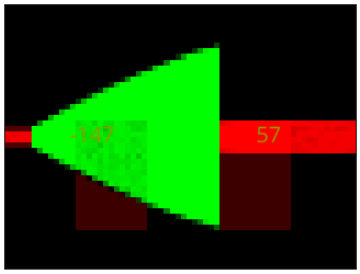
Répondez aux questions suivantes :

1. Si le robot se trouve dans la configuration suivante, quelle action va-t-il effectuer (répondez pour chacun des deux entraînements) ?



À gauche on lit "-98" et à droite "-135"

2. Si le robot se trouve dans la configuration suivante, quelle action va-t-il effectuer (répondez pour les deux entraînements) ?



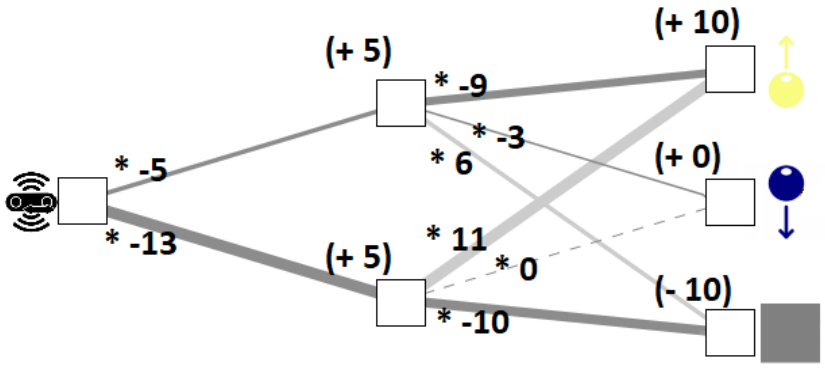
À gauche on lit “-147” et à droite “57”

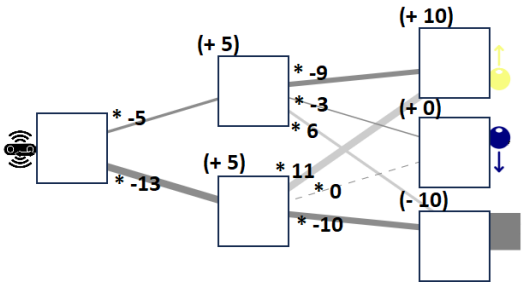
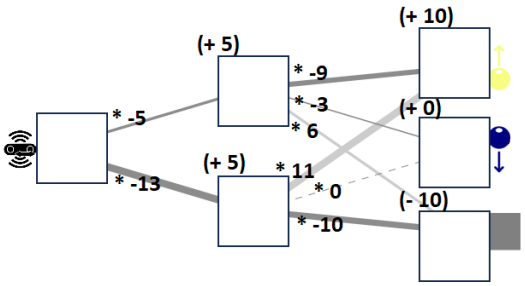
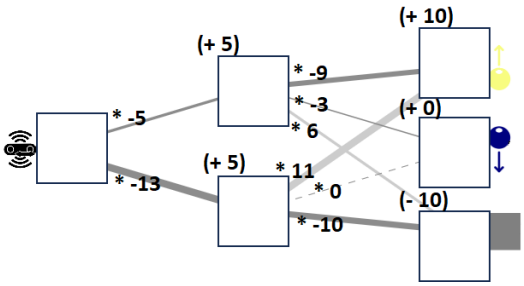
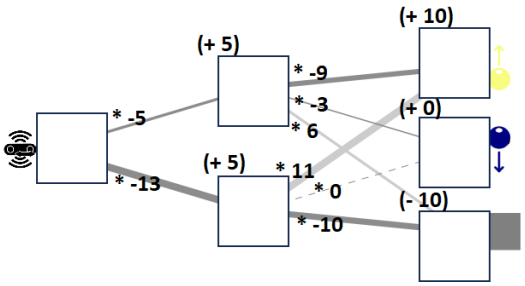
3. Quel entraînement est le plus performant pour que le robot se déplace autour de l’arène dans le sens d'une aiguille d'une montre (Justifier)?

4. Quels algorithmes d’apprentissage ont été utilisés pour obtenir ces deux espaces d'états ? Quels sont les indices qui vous permettent de le déduire ?

Exercice 4 : Réseau de neurones

Pour le réseau de neurones ci-dessous, calculez les activations des neurones de la couche intermédiaire et des neurones de sortie, avec les fonctions d'activation et différentes valeurs d'entrée ci-dessous. **Rappel :** la fonction d'activation va s'appliquer sur les deux neurones de la couche intermédiaire, mais pas sur la couche de sortie.



	Fonction d'activation **Leaky ReLU** $y = \begin{cases} \frac{z}{10} \; si \; z < 0 \\ z \; si \; z \geq 0 \end{cases}$	Fonction d'activation **Heaviside** $y = \begin{cases} 0 \; si \; z < 0 \\ 1 \; si \; z \geq 0 \end{cases}$
Valeur d'input : 0		
Valeur d'input : -2		

Exercice 5 : Apprentissage du morpion en Q-learning

Nous allons utiliser le Q-learning pour trouver la stratégie optimale pour jouer au morpion contre un joueur “moyen”. Mais commençons par quelques questions de cours.

Question 1 : à propos de la formule ci-contre. $target = (1-\gamma)r + \gamma \max_a Q(s', a)$

Comment appelle-t-on le paramètre γ ?

Quelles sont ses bornes (valeurs minimales et maximales) ?

Question 2 : à propos de la formule ci-contre. $Q'(s,a) = (1-\alpha)Q(s,a) + \alpha\ target$

Comment appelle-t-on le paramètre α ?

Quelles sont ses bornes (valeurs minimales et maximales) ?

Le jeu du morpion (tic-tac-toe) est un jeu à deux joueurs qui se pratique sur une grille de taille 3x3. Les joueurs placent chacun à leur tour un symbole dans une case inoccupée. Le jeu se termine lorsqu’un joueur parvient à aligner 3 de ses symboles en ligne ou en diagonale, ce joueur est alors vainqueur ; ou lorsque toute les cases du plateau sont occupées sans qu’un joueur ait réussi à aligner 3 symboles : le résultat est alors un match nul.

La figure ci-dessous (page 10) est un arbre représentant différentes possibilités de déroulement d'une partie de morpion (en se restreignant aux coups les plus pertinents, sinon l'arbre de tous les coups possibles serait trop gros). Le premier joueur sera l'IA, elle utilise le symbole X ; le second joueur utilise le symbole O. Nous allons supposer que ce second joueur (symbole O) joue de manière aléatoire entre les coups représentés dans l'arbre : à partir d'une position donnée, il choisit de manière équiprobable un coup parmi les possibilités représentées. Les probabilités de ces choix ont été représentées en vert sur la figure.

L'objectif de l'exercice est de déterminer la meilleure stratégie pour l'IA grâce à l'algorithme du Q-learning. Nous allons donc fixer une récompense de 100 en cas de victoire de la joueuse X, une récompense négative de -100 en cas de victoire du joueur O, et une récompense nulle en cas de match nul. **Attention :** la récompense n'est gagnée que lorsque le jeu est terminé, les coups joués en cours de partie ne rapportent pas de récompense immédiate !

Plutôt que d'appliquer l'algorithme du Q-learning étape par étape, nous pourrons déterminer directement vers quelles valeurs limites l'algorithme va converger, en utilisant l'équation de Bellman. En effet, au terme de l'apprentissage les Q-valeurs vérifient l'équation de Bellman : pour tout état s et action a, la Q-valeur $Q(s,\ a)$ vérifie :

$$Q(s,\ a)\ =\ E[r\ +\ \gamma\ max_{a'}\ Q(s',\ a')],$$

où E représente l'espérance mathématique (c'est à dire la moyenne en probabilité), s' le nouvel état atteint et $max_{a'}\ Q(s',\ a')$ la meilleure Q-valeur possible à partir de ce nouvel état.

Nous utiliserons γ = 1, ce qui est le plus approprié dans la résolution d'un jeu.

Question 3 :

Que représente l'*état s* dans le cadre du jeu du morpion ?

Question 4 :

Que représente l'*action a* dans le cadre du jeu du morpion ?

Question 5 :

Que représente la notion d'*environnement* en apprentissage par renforcement ? Et en quoi consiste l'environnement ici dans le cadre du jeu du morpion ?

Question 6 :

Dans la figure page 10, nous avons complété les Q-valeurs de trois couples (s, a), à droite et vers le haut du graphe :

- dans un des trois cas, le coup donne lieu de manière univoque à un nul, c'est à dire récompense 0 ; on a donc

  $$Q(s,\ a)\ =\ E[r\ +\ \gamma\ max_{a'}\ Q(s',\ a')]\ =\ E[r]\ =\ 0$$

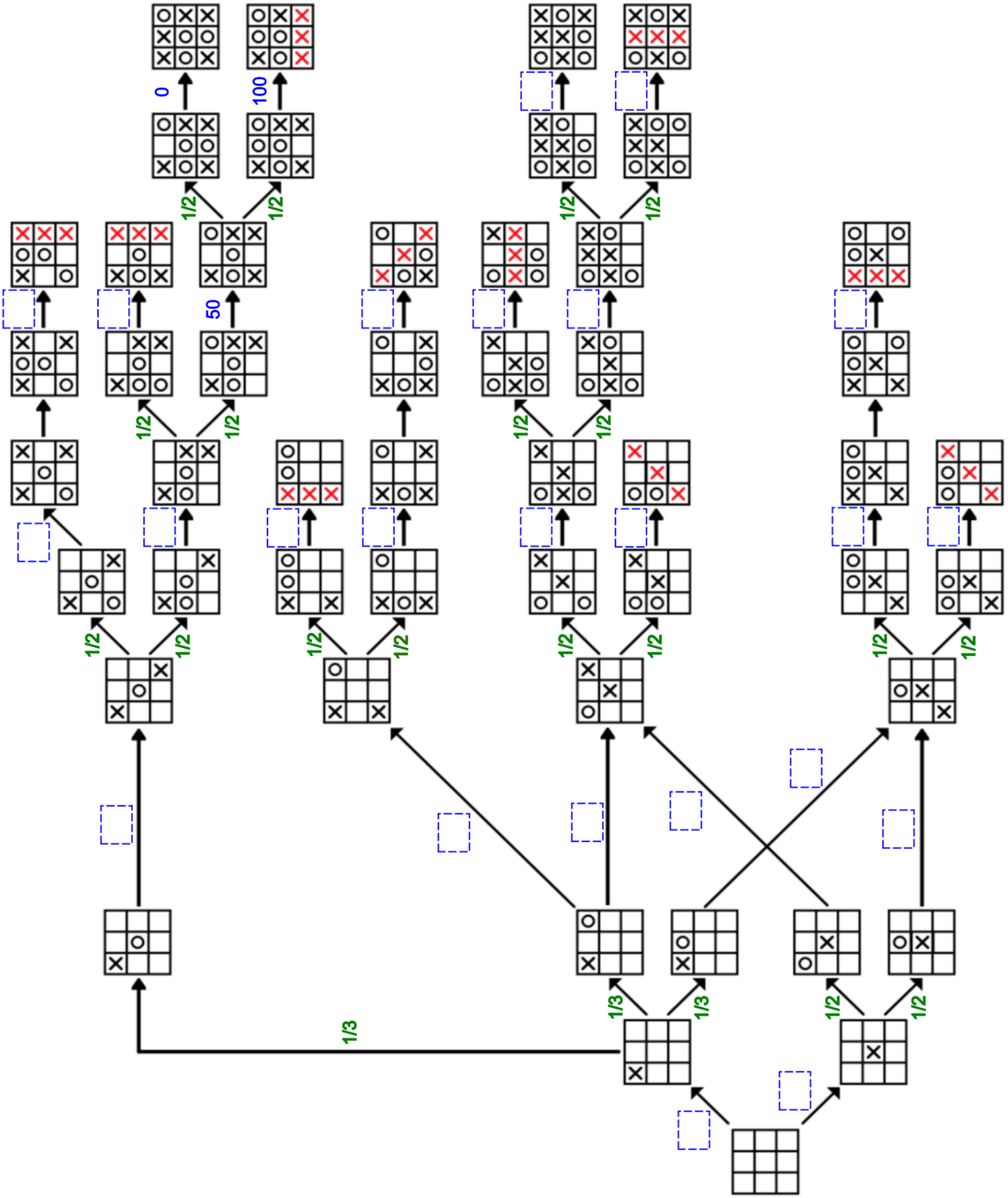
  (noter que comme la partie est finie, la partie $\gamma\ max_{a'}\ Q(s',\ a')$ est nulle)

- de même un deuxième cas conduit à la victoire :

  $$Q(s,\ a)\ =\ E[r\ +\ \gamma\ max_{a'}\ Q(s',\ a')]\ =\ E[r]\ =\ 100$$

- dans un troisième cas, on joue un coup tel que, ensuite, le deuxième joueur va jouer en bas avec probabilité ½ (récompense immédiate $r\ =\ 0$, nouvel état s' tel que $max_{a'}\ Q(s',\ a')\ =\ 0$), et jouer en haut avec probabilité ½ ($r\ =\ 0$, nouvel état s' tel que $max_{a'}\ Q(s',\ a')\ =\ 0$) ; on a donc (rappelons que $\gamma\ =\ 1$)

  $$Q(s,\ a)\ =\ E[r\ +\ \gamma\ max_{a'}\ Q(s',\ a')]\ =\ 1/2\ *\ (0\ +\ 0)\ +\ 1/2\ *\ (0\ +\ 100)\ =\ 50$$


0
100
50
1/2
1/3

Compléter les Q-valeurs limites à l'aide de l'équation de Bellman. Les emplacements à compléter sont indiqués par des rectangles bleus. Vous remarquerez qu'il faut commencer par remplir les valeurs les plus à droite.

On pourra arrondir les valeurs à l'unité la plus proche.

Question 7 :

Dans la configuration étudiée ici, et en supposant que la première joueuse choisit toujours l'action avec la Q-valeur la plus élevée, quelle est la probabilité de victoire pour la première joueuse (symbole X) ?

Avec les mêmes hypothèses, quelle est la probabilité d'un match nul ?

Exercice 6 : Étude de document

Pour cet exercice, veuillez lire les textes et entourer/surligner la ou les bonne(s) réponse(s) ci-dessous.

Images obtenues en utilisant DALL-E 2

DALL-E est un générateur d'images par IA conçu par OpenAI, connu pour avoir développé ChatGPT. Il permet de créer des visuels de toutes sortes à partir d'un prompt textuel. Son nom est un mot-valise évoquant à la fois le robot de Pixar WALL-E et le peintre Salvador Dalí. DALL-E est une version de 12 milliards de paramètres de GPT-3, entraînée pour générer des images à partir de descriptions textuelles, en utilisant un ensemble de données d'image annotées.

Q1. Qu'est-ce que DALL-E ?

a. Un modèle de traitement du langage naturel.
b. Un modèle d'IA générative pour la création d'images.
c. Un algorithme de reconnaissance d'objets.

Q2. Sur quoi DALL-E a-t-il été formé pour générer des images ?

a. Uniquement des descriptions textuelles.
b. Des paires texte-image.
c. Des images uniquement.

Q3. D'après ce qu'il est décrit au-dessus, est-ce que DALL-E ?

a. Est basée sur un apprentissage non-supervisé.
b. Est basée sur un apprentissage par renforcement.
c. Est basée sur un apprentissage supervisé.

Le fonctionnement de DALL-E repose sur une architecture de réseau de neurones appelée générateur d'images, qui prend en entrée une description textuelle et produit une image correspondante. Voici les étapes clés du processus :

- Pré-entraînement : DALL-E est initialement pré-entraîné sur un large ensemble de données contenant des paires texte-image. Ce pré-entraînement permet au modèle d'apprendre la structure sous-jacente des données visuelles et textuelles.
- Encodage de la description : Lorsqu'une description textuelle est fournie en entrée, le modèle utilise un encodeur pour convertir cette description en une représentation vectorielle, également appelée espace latent. Cette représentation capture les caractéristiques sémantiques de la description.

- Décodeur d'images : Le vecteur latent est ensuite fourni au décodeur d'images, qui génère une image correspondante. Ce décodeur est capable de transformer la représentation vectorielle en une image réaliste et cohérente en utilisant les connaissances acquises lors du pré-entraînement.
- Optimisation : Le modèle est optimisé pour minimiser la divergence entre l'image générée et les exemples d'entraînement réels. Cela permet d'affiner les paramètres du générateur afin d'améliorer la qualité des images générées.

Q4. Qu'est-ce que l'espace latent représente dans le contexte de DALL-E ?

a. L'espace des représentations textuelles.
b. La structure sous-jacente des données visuelles.
c. La dimension des images générées.

Q5. Comment le décodeur d'images utilise-t-il le vecteur latent pour générer une image correspondante ?

a. En minimisant la divergence.
b. En affinant les paramètres du générateur.
c. En utilisant les connaissances acquises lors du pré-entraînement.

Q6. Pourquoi DALL-E est-il soumis à une phase d'optimisation ?

a. Pour maximiser la divergence entre l'image générée et les exemples d'entraînement réels.
b. Pour minimiser la divergence entre l'image générée et les exemples d'entraînement réels.
c. Pour ajuster l'encodeur du modèle.

Afin de rendre DALL-E accessible à un large public, OpenAI a pris des mesures pour réduire les risques liés à ces modèles puissants de génération d'images. Des garde-fous ont été mis en place pour éviter que les images générées enfreignent la politique de contenu d'OpenAI. DALL-E est formé sur des centaines de millions d'images légendées provenant d'Internet, et OpenAI supprime et réévalue certaines de ces images pour influencer l'apprentissage du modèle. Cette stratégie vise notamment à filtrer les images violentes et sexuelles du jeu de données d'entraînement de DALL-E. Cette étape

cruciale a pour objectif d'empêcher le modèle d'apprendre à générer du contenu graphique ou explicite en réponse aux requêtes des utilisateurs, évitant ainsi la production involontaire d'images inappropriées. Cependant, cette démarche peut entraîner des biais potentiels, car les modèles formés sur des données filtrées ont tendance à amplifier certaines préférences, comme générer plus d'images mettant en scène des hommes et moins d'images représentant des femmes par rapport aux modèles formés sur l'ensemble de données original et non filtré.

Q7. Quelle est la principale raison pour laquelle OpenAI a mis en place des garde-fous lors de la mise à disposition de DALL-E au public ?

a. Pour empêcher les utilisateurs d'accéder aux fonctionnalités avancées de DALL-E.
b. Pour éviter que les images générées enfreignent la politique de contenu d'OpenAI.
c. Afin de limiter la création d'images susceptibles d'être utilisées à des fins non éthiques ou de causer du tort à autrui.

Q8. Quelle est la principale raison pour laquelle OpenAI supprime et réévalue certaines images du jeu de données d'entraînement de DALL-E ?

a. Pour réduire la taille du jeu de données.
b. Pour influencer délibérément l'apprentissage du modèle.
c. Pour accélérer le processus d'entraînement de DALL-E.

Sources :

- https://www.assemblyai.com/blog/how-dall-e-2-actually-works/#what-is-dall-e-
- https://openai.com/blog