

Découverte de l'Intelligence Artificielle

Evaluation écrite

5 janvier 2023

Thomas Deneux, thomas.deneux@cnrs.fr

Numéro de copie anonymisée :

1. Définissez les termes suivants en 1 à 3 lignes chacun. Expliquez comment chacun d'eux est lié aux autres (par exemple, un terme est un sous-cas d'un autre terme, etc.)

- Apprentissage supervisé

- Apprentissage non supervisé

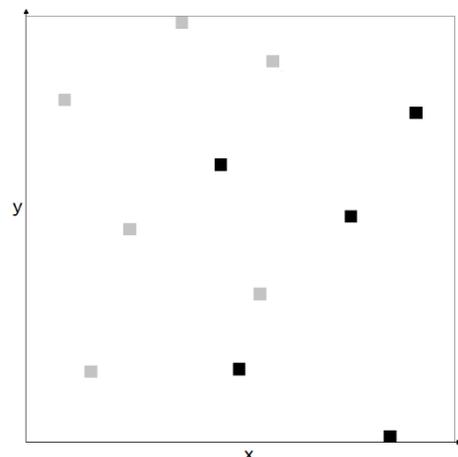
- Apprentissage par renforcement

- Apprentissage profond (« deep learning »)

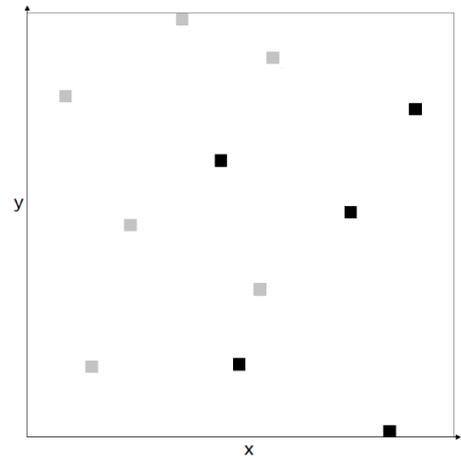
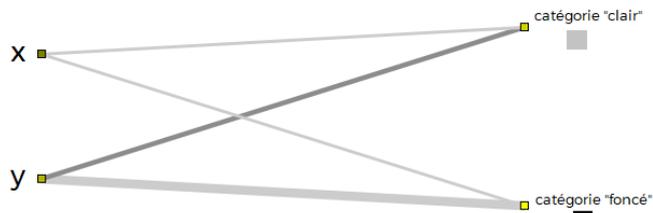
- Machine learning

2. Voici ci-dessous un ensemble de 11 points dans l'espace à deux dimensions, appartenant à 2 catégories (6 points clairs contre 5 points foncés). Supposons qu'on utilise ces points pour entraîner différents algorithmes d'apprentissage supervisé à classifier n'importe quel point du plan entre ces 2 catégories. Tracez approximativement dans le graphe la ligne de démarcation entre les catégories qui sera obtenue :

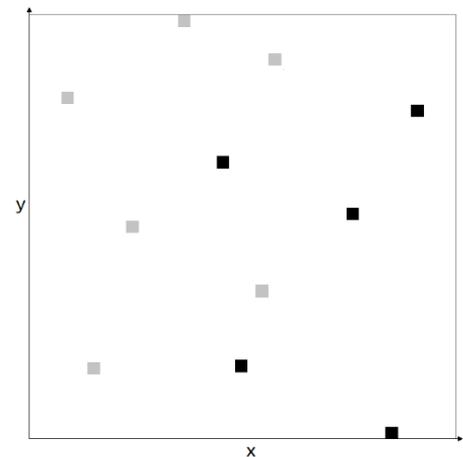
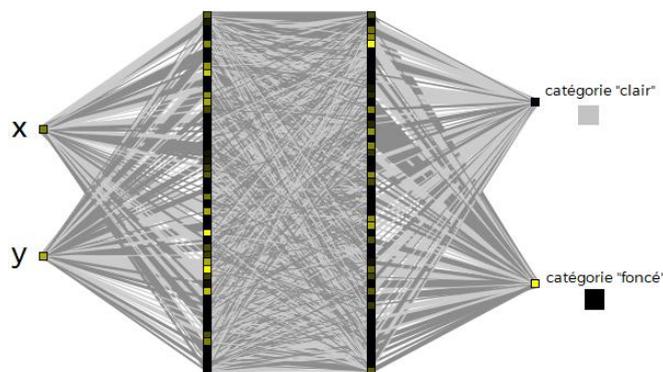
- Si on utilise l'algorithme des K plus proches voisins (avec $K = 1$)



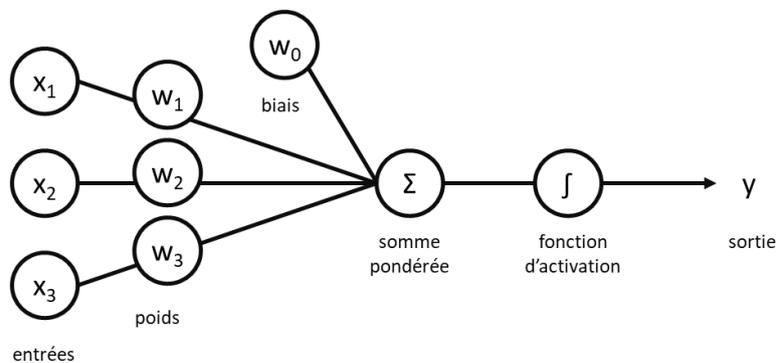
- Si on utilise un réseau de neurones à une seule couche



- Si on utilise un réseau de neurones plus complexe (par exemple 2 couches de 50 neurones)



3. L'image ci-dessous représente les opérations effectuées par un seul neurone artificiel. Écrivez ces opérations exactement en symboles mathématiques.



C'est-à-dire :

- Notez z le résultat de la somme pondérée et écrivez la valeur de z en fonction des entrées, poids et biais
- Choisissez un exemple de fonction d'activation pour écrire y en fonction de z

4. Quelle serait la sortie du code Python suivant ?

```
x = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(x[1])
print(x[1][2])
```

5. Analyse de document : ChatGPT

Le 30 novembre 2022, la fondation OpenAI a mis en ligne un chatbot conversationnel nommé « ChatGPT », capable de discussions époustouflantes. (Je vous encourage de retour chez vous à l'essayer : Cherchez sur Internet « ChatGPT » ; vous pourrez également essayer « Dall-e » qui génère des images automatiquement à partir de descriptions textuelles)

Vous trouverez ci-joint la page de blog <https://openai.com/blog/chatgpt/> et sa traduction et légère simplification en français. Nous allons voir comment, suite au cours Découverte de l'IA, vous êtes capable de comprendre les quelques descriptions techniques sur comment a été entraîné ChatGPT ! Comme vous le verrez en effet ces explications exigent déjà un certain niveau de connaissance pour pouvoir être comprises, elles ne disent pas forcément tout, et sont certainement seulement une simplification du vrai processus ; les questions ci-dessous vont vous guider.

Cochez les bonnes réponses ci-dessous. Pour certaines questions il faut cocher plusieurs bonnes réponses.

L'article parle beaucoup de « modèles ». Un modèle en Machine Learning est comme une « fonction programmable ». Par exemple, un réseau de neurones est un modèle. Un modèle génère des sorties en fonction d'entrées qu'on lui donne. Un modèle a des paramètres qui peuvent être modifiés par un algorithme d'apprentissage. Dans le cours nous avons parlé de classifieur plutôt que de modèle : un classifieur est une catégorie particulière de modèle qui comme son nom l'indique, fait de la classification (ses sorties sont des numéros de catégorie).

Etant donné cette définition d'un modèle en Machine Learning, est-ce que :

- l'algorithme des K plus proches voisins est un modèle
- un réseau de neurone est un modèle
- la table du Q-learning est un modèle

ChatGPT est :

- une IA
- un agent conversationnel (chatbot)
- un modèle de Machine Learning

ChatGPT est capable de :

- répondre à des questions de culture générale
- inventer des poèmes
- se rappeler des précédentes questions qu'on lui a posées et de ses précédentes réponses

Qu'est-ce que le modèle ChatGPT prend en entrée ?

- la dernière phrase tapée par l'utilisateur, mais également l'historique des questions et réponses précédentes
- la dernière phrase tapée par l'utilisateur uniquement
- une très grande base de donnée de questions et de réponses

Qu'est-ce que le modèle ChatGPT renvoie comme sorties ?

- des réponses sous forme d'un texte d'un ou plusieurs paragraphes
- des nombres

Contrairement aux exemples d'apprentissages vus en cours, ChatGPT n'a pas été entraîné en utilisant un seul algorithme et une seule base de données, mais selon une procédure complexe en plusieurs étapes, et mettant en jeu plusieurs algorithmes et plusieurs bases de données. Il est une évolution du modèle GPT-3.5 qui a été entraîné auparavant. Le texte parle également d'un autre modèle appelé InstructGPT. Est-ce que :

- ChatGPT est une évolution de InstructGPT, qui est lui-même une évolution de GPT-3.5?
- ou ChatGPT et InstructGPT sont tous les deux des évolutions de GPT-3.5, mais aucun n'est une évolution de l'autre ?

Dans le cadre des réseaux de neurones vus en cours pour piloter le robot AlphaI, serait-il envisageable d'entraîner un tel réseau avec de l'apprentissage supervisé, puis de poursuivre cet entraînement avec de l'apprentissage par renforcement ?

- oui : les connexions dans le réseau seront modifiées tout d'abord par l'apprentissage supervisé, puis par l'algorithme d'apprentissage par renforcement (par exemple par DQN étudié en cours)
- non : au démarrage de l'apprentissage par renforcement les connexions sont forcément réinitialisées aléatoirement et on ne peut pas garder de trace d'un premier apprentissage supervisé

Le texte joint n'explique pas comment GPT-3.5 a été entraîné, mais on trouve ailleurs sur Internet que GPT-3.5 a été entraîné pour compléter automatiquement le début d'un texte. A votre avis, grâce à quel type de base de donnée GPT-3.5 a été entraîné ?

- à partir d'un grand nombre de textes d'exemples écrits par des entraîneurs humains spécialement pour entraîner GPT-3.5
- à partir d'un maximum de textes déjà disponibles (Wikipedia, encyclopédies, livres scannés, sites Internet, etc.)

L'étape 1 décrite dans le texte et dans la figure est une première étape d'amélioration de GPT-3.5, en quoi consiste-t-elle ?

- un apprentissage supervisé utilisant comme données des exemples de réponses fournis par un entraîneur humain
- un apprentissage supervisé utilisant comme données des réponses générées par GPT-3.5

Comment fonctionnent les étapes 2 et 3 ?

- dans l'étape 2, la première version de ChatGPT issue de l'étape 1 est améliorée avec de l'apprentissage par renforcement (c'est un humain qui donne des récompenses) ; dans l'étape 3, elle est encore améliorée, toujours avec de l'apprentissage par renforcement (mais cette fois les récompenses sont générées de manière automatique)
- dans l'étape 2, un nouveau modèle auxiliaire est entraîné à donner des notes aux réponses de ChatGPT ; dans l'étape 3, la première version de ChatGPT issue de l'étape 1 est améliorée avec de l'apprentissage par renforcement (les récompenses sont déterminées par le modèle auxiliaire de l'étape 2)

6. Apprentissage d'un labyrinthe avec le Q-learning

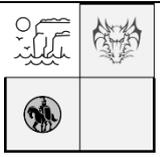
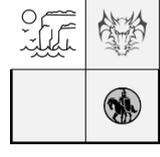
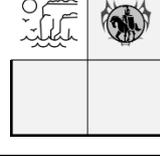
Un chevalier se trouve dans un labyrinthe de 3 cases. Il se déplace en choisissant une des quatre directions cardinales. Vous allez exécuter à la main toutes les étapes de calcul de l'algorithme Q-learning au cours d'un apprentissage du déplacement dans ce labyrinthe !



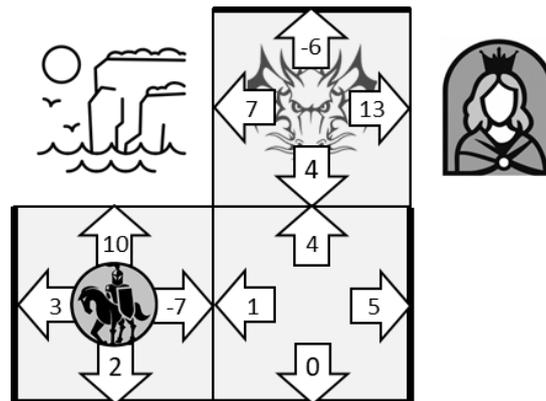
Voici comment fonctionnent les déplacements et les récompenses :

- si le chevalier essaie de franchir un mur épais, il reste dans la même case et n'obtient aucune récompense.
- s'il se déplace vers la princesse, il est récompensé de +50 points et une nouvelle partie recommence (il est placé sur la case en bas à gauche).
- s'il se déplace vers la falaise, il meurt, perd -10 points, et une nouvelle partie recommence.
- s'il se déplace vers la case du dragon un combat s'engage : avec une chance sur deux le chevalier reste vivant, n'obtient aucune récompense et reste sur la case ; avec une chance sur deux il meurt, perd -10 points, et la partie recommence.

La table du Q-learning récapitule les « valeurs » assignées à chacune des 4 actions à partir de chacun des 3 états possibles. On l'initialise aléatoirement (avec des valeurs qui ne sont donc pas correctes).

« Q-valeur » de l'action a depuis l'état s	Action 	Action 	Action 	Action 
Etat 	10	-7	2	3
Etat 	4	5	0	1
Etat 	-6	13	4	7

Nous allons représenter cette table sous la forme plus agréable suivante :



La partie commence : l'IA choisit l'action de plus grande valeur d'après sa table :  . Le chevalier tombe de la falaise et reçoit la récompense $R = -10$, la partie s'arrête. Nous utilisons la formule du Q-learning pour mettre à jour la valeur $Q(s_t, a_t)$ de l'action qui vient d'être effectuée :

$$target = R_t + \gamma \max_a Q(s_{t+1}, a)$$

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha target$$

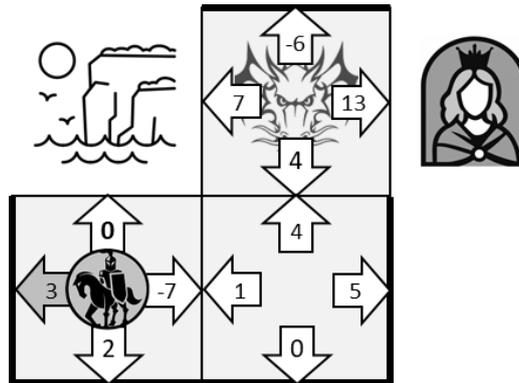
Pour simplifier nous choisissons comme *vitesse d'apprentissage* $\alpha = 0.5$ et *facteur d'actualisation* $\gamma = 0.5$, et lors des calculs nous ne garderons pas les nombres après la virgule, mais arrondirons à l'entier le plus proche (si on a .5 après la virgule, on arrondit dans la direction de zéro, par exemple $-3.5 \rightarrow -3$).

Comme la partie s'arrête avec la mort du chevalier, il n'y a pas « d'état suivant » s_{t+1} , donc pour cette fois-ci la partie $\gamma \max_a Q(s_{t+1}, a)$ de l'équation est ignorée :

$$target = R_t = -10$$

$$Q(s_t, a_t) \leftarrow (1 - \alpha) * Q(s_t, a_t) + \alpha * target = 0.5 * 10 + 0.5 * (-10) = 0$$

La Q-valeur est donc remplacée par 0. Une nouvelle partie recommence, l'IA choisit à nouveau l'action de plus grande valeur, mais à présent c'est la direction ouest .



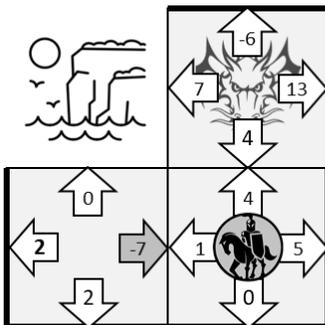
Le chevalier reçoit une récompense de 0 et reste sur la même case. Nous calculons la mise à jour de la Q-valeur :

$$target = R_t + \gamma \max_a Q(s_{t+1}, a) = 0 + 0.5 * 3 = 1.5$$

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha target = 0.5 * 3 + 0.5 * 1.5 = 1.5 + 0.75 = 2.25$$

On arrondit à 2.

A présent l'IA fait faire une *exploration* au chevalier en choisissant l'action (non préférée) .



Récompense reçue : 0.

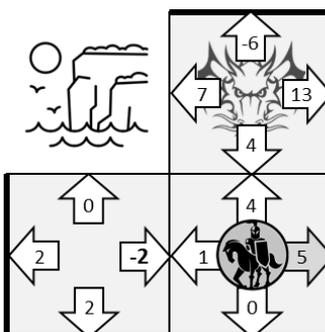
Calcul de la mise à jour :

$$target = R_t + \gamma \max_a Q(s_{t+1}, a) = 0 + 0.5 * 5 = 2.5$$

$$Q(s_t, a_t) \leftarrow 0.5 * (-7) + 0.5 * 2.5 = -3.5 + 1.25 = -2.25$$

On arrondit à -2.

Nous continuons :



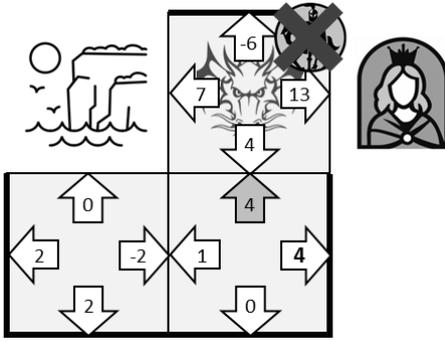
L'IA choisit ensuite l'action de plus forte valeur .

Récompense reçue : 0

$$target = R_t + \gamma \max_a Q(s_{t+1}, a) = 0 + 0.5 * 5 = 2.5$$

$$Q(s_t, a_t) \leftarrow 0.5 * 5 + 0.5 * 2.5 = 2.5 + 1.25 = 3.75$$

On arrondit à 4.



Action choisie :



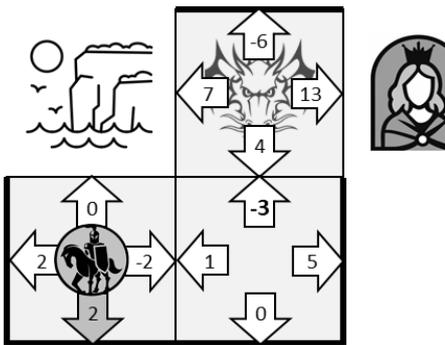
Le chevalier arrive sur la case du dragon, l'issue du combat lui est fatale, il meurt et perd -10 points. Fin de partie.

$$target = R_t = -10$$

$$Q(s_t, a_t) \leftarrow 0.5 * 4 + 0.5 * (-10) = 2 - 5 = -3$$

Le chevalier repart de la case sud-ouest.

A vous de faire les calculs maintenant !



Action choisie :



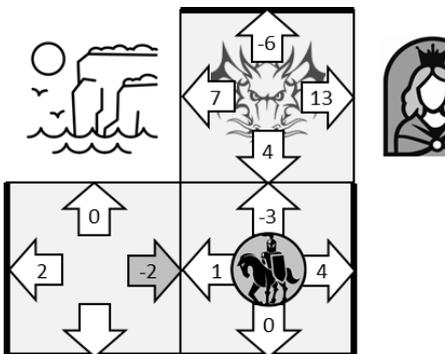
Le chevalier reste sur la même case, récompense : 0

$$target = R_t + \gamma \max_a Q(s_{t+1}, a) =$$

$$Q(s_t, a_t) \leftarrow 0.5 * Q(s_t, a_t) + 0.5 * target =$$

On arrondit à

Désormais, ajoutez les valeurs manquantes dans les flèches suite aux apprentissages



Action choisie :



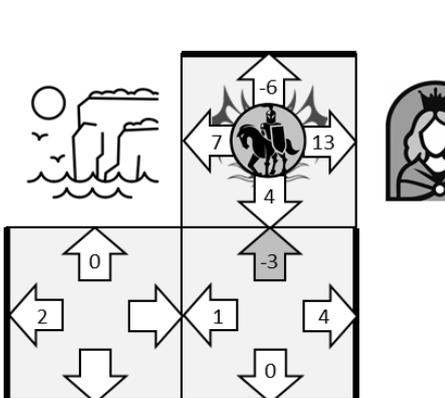
(exploration)

Récompense : 0

$$target =$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à



Action choisie :



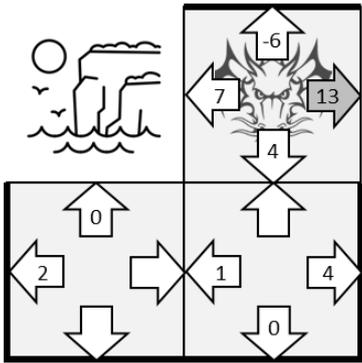
(exploration ; oui il y a beaucoup d'explorations heureuses, sinon ce serait trop long !)

Nouveau combat contre le dragon, cette fois la chance sourit au chevalier, il reste en vie et reste sur la case. Récompense : 0.

$$target =$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à



Action choisie :

13

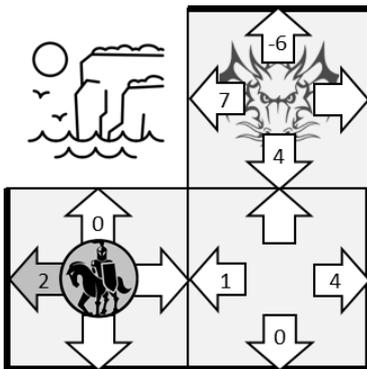
Le chevalier a rejoint la princesse ! Fin de partie. Récompense : +50. Comme d'habitude, on met à jour la valeur de l'action :

$$target = R_t = +50$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à

Une nouvelle partie commence sur la case sud-ouest.



Action choisie :

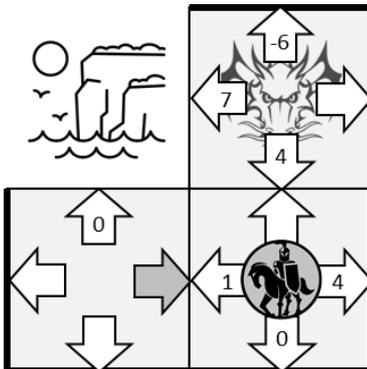
2

Le chevalier reste sur la même case. Récompense : 0.

$$target =$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à



Action choisie :



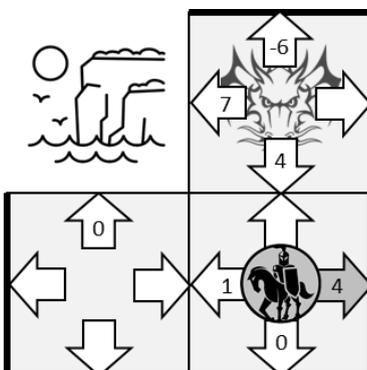
(encore une exploration heureuse)

Récompense : 0

$$target =$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à



Action choisie :

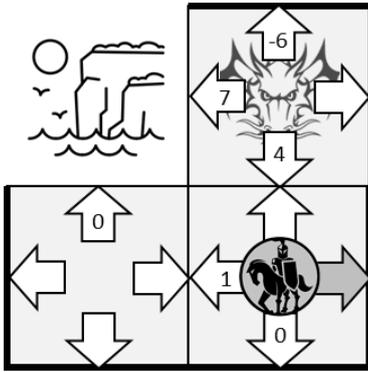
4

Récompense : 0. On reste sur la même case.

$$target =$$

$$Q(s_t, a_t) \leftarrow$$

On arrondit à



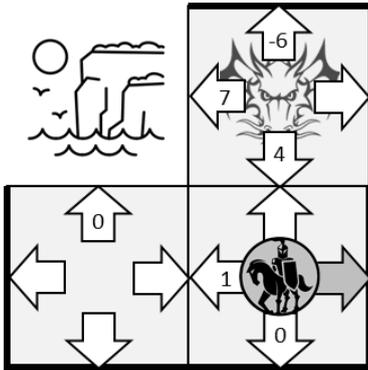
Même action choisie : 

Récompense : 0. On reste sur la même case.

$target =$

$Q(s_t, a_t) \leftarrow$

On arrondit à



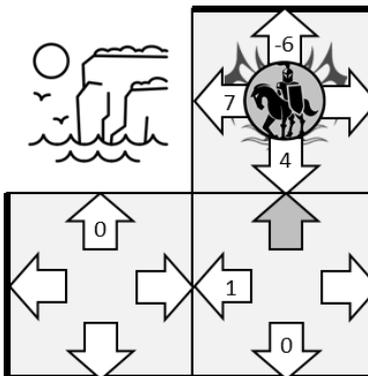
Même action choisie : 

Récompense : 0. On reste sur la même case.

$target =$

$Q(s_t, a_t) \leftarrow$

On arrondit à



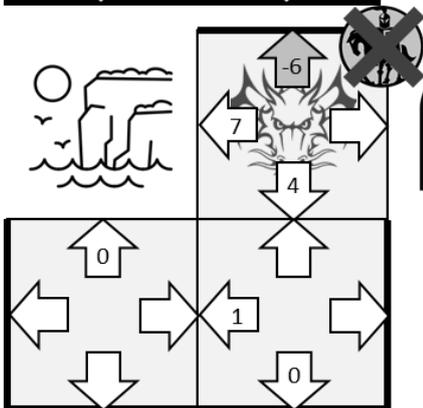
Action choisie : 

Nouveau combat contre le dragon. Le chevalier résiste à nouveau.
Récompense : 0.

$target =$

$Q(s_t, a_t) \leftarrow$

On arrondit à



Action choisie :  (exploration... non favorable cette fois)

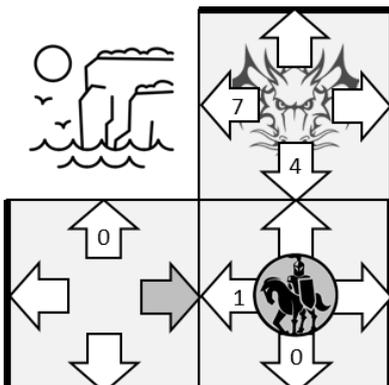
Le chevalier reste sur la même case, doit combattre à nouveau le dragon, et cette fois meurt. Récompense : -10. Fin de partie.

$target = R_t = -10$

$Q(s_t, a_t) \leftarrow$

On arrondit à

Le chevalier repart de la case sud-ouest.



Action choisie : 

Récompense : 0.

$target =$

$Q(s_t, a_t) \leftarrow$

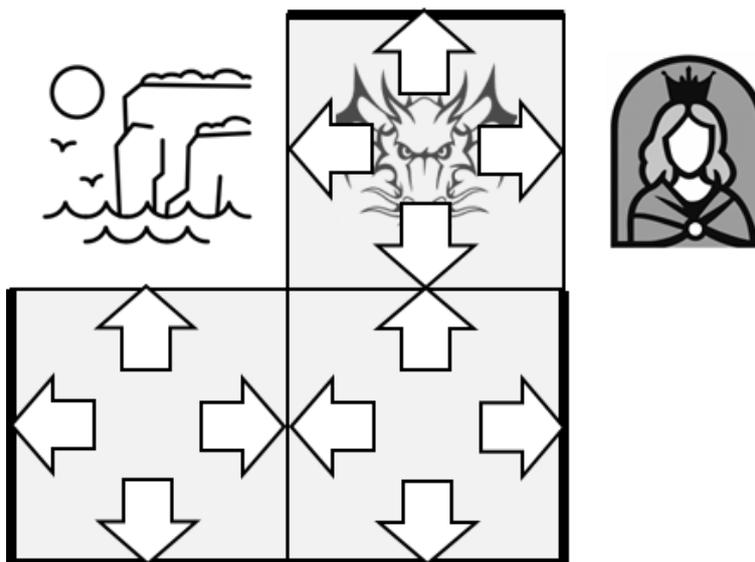
On arrondit à

Désormais, les Q-valeurs estimées reflètent bien mieux la vraie valeur de chaque action, et en particulier si le chevalier ne fait plus d'exploration, il ira toujours à la princesse en partant de la case sud-ouest (pourvu qu'il ne soit pas arrêté par le dragon).

7. Question facultative (plus difficile) : convergence de l'apprentissage du labyrinthe

Si on continue l'apprentissage, les Q-valeurs vont osciller, notamment celles des actions qui arrivent sur la case dragon, car le calcul de *target* dépend de l'issue aléatoire du combat entre le chevalier et le dragon.

Mais si on diminue la vitesse d'apprentissage α , ces oscillations pourront être petites, et les valeurs de la table Q finiront par converger au bout d'un grand nombre d'actions et de mises à jour. Seriez-vous capable de déterminer les valeurs au terme de cette convergence ? Notez les ci-dessous (on garde $\gamma = 0.5$; cette fois, déterminez les valeurs exactes, sans arrondir à l'entier le plus proche):



Note : au terme de la convergence, les Q-valeurs vérifient l'équation de Bellman :

$$Q(s_t, a_t) = E(R_t + \gamma \max_a Q(s_{t+1}, a)),$$

où E désigne l'espérance mathématique.



ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

[TRY CHATGPT ↗](#)

November 30, 2022 ; 13 minute read

We are excited to introduce ChatGPT to get users' feedback and learn about its strengths and weaknesses. During the research preview, usage of ChatGPT is free. Try it now at chat.openai.com.

Samples

In the following sample, ChatGPT is able to understand the reference ("it") to the subject of the previous question ("fermat's little theorem").

User: what is the fermat's little theorem

ChatGPT: Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p , then a^p (a to the power p) is congruent to a modulo p . In other words, if we divide a^p by p , the remainder is always a .

For example, if $p=5$ and $a=2$, then $2^5=32$, and 32 divided by 5 has a remainder of 2. Therefore, 2^5 is congruent to 2 modulo 5, and Fermat's Little Theorem holds for this case.

Fermat's Little Theorem is often used in cryptography and other applications where it is necessary to perform modular arithmetic operations quickly and efficiently. It is also a useful tool for proving other theorems in number theory

User: how is it used in cryptography?

ChatGPT: Fermat's Little Theorem is used in cryptography in several ways. One of the most common applications is in the generation of so-called "public-key" cryptography systems, which are used to securely transmit messages over the internet and other networks.

In a public-key cryptography system, each user has a pair of keys: a public key, which is widely known and can be used by anyone to encrypt a message intended for that user, and a private key, which is known only to the user and is used to decrypt messages that have been encrypted with the corresponding public key.

One way to generate these keys is to use prime numbers and Fermat's Little Theorem. For example, suppose we want to generate a public-key cryptography system for a user with the initials "ABC". We might choose two large prime numbers, p and q , and then compute the product $n = pq$.

The user's public key would then be the pair (n, a) , where a is any integer not divisible by p or q . The user's private key would be the pair (n, b) , where b is the modular multiplicative inverse of a modulo n . This means that when we multiply a and b together, the result is congruent to 1 modulo n .

To encrypt a message with the user's public key (n, a) , we first convert the message into a number m (using some agreed-upon scheme), and then compute the encrypted message c as $c = m^a \pmod n$. To decrypt the message, the recipient simply computes $m = c^b \pmod n$, which (by Fermat's Little Theorem) is equivalent to $m = (m^a)^b \pmod n = m^{ab} \pmod n = m \pmod n$.

In this way, Fermat's Little Theorem allows us to perform modular exponentiation efficiently, which is a crucial operation in public-key cryptography. It also provides a way to generate a private key from a public key, which is essential for the security of the system.

User: write a limerick about it



ChatGPT: There once was a theorem by Fermat

That many found quite elegant
It said if you take
A prime number and make
The power of a not divisible by it
The result, congruent, will fit
In cryptography, it's quite handy
For encrypting, it's dandy
Fermat's Little Theorem, oh so neat
In number theory, it can't be beat!

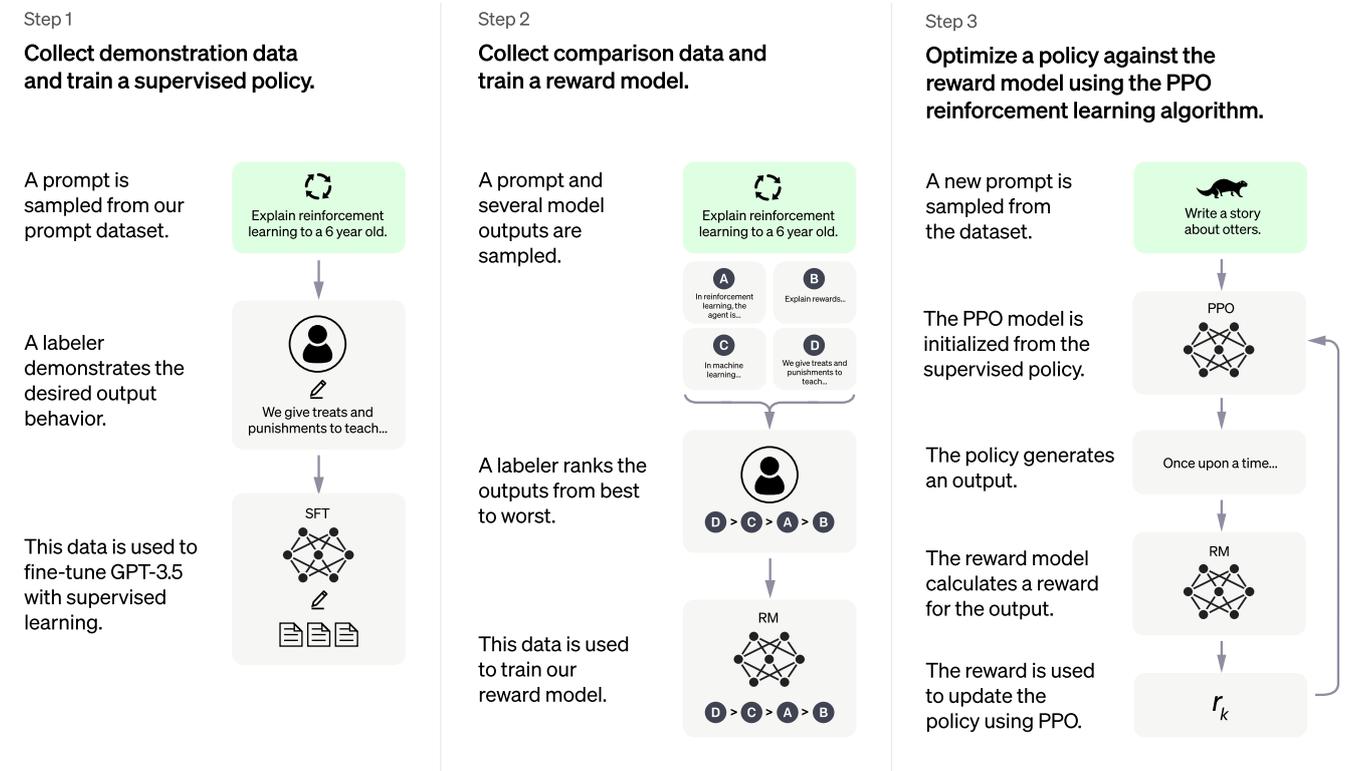
User: summarize the conversation so far

ChatGPT: Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p , then a^p (a to the power p) is congruent to a modulo p . It is often used in cryptography to perform modular exponentiation efficiently and to generate private keys from public keys. It is also a useful tool in number theory for proving other theorems. I also wrote a limerick about Fermat's Little Theorem.

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.



ChatGPT is fine-tuned from a model in the GPT-3.5 series, which finished training in early 2022. You can learn more about the 3.5 series here. ChatGPT and GPT 3.5 were trained on an Azure AI supercomputing infrastructure.



ChatGPT : Optimisation de modèles linguistiques pour le dialogue

Nous avons entraîné un modèle appelé ChatGPT qui interagit de manière conversationnelle. Le format de dialogue permet à ChatGPT de répondre à des questions suivies, d'admettre ses erreurs, de contester des prémisses incorrectes et de rejeter des demandes inappropriées. ChatGPT est un modèle cousin de InstructGPT, qui est entraîné à suivre une instruction dans un prompteur et à fournir une réponse détaillée.

Nous sommes ravis de présenter ChatGPT afin de recueillir les réactions des utilisateurs et de connaître ses forces et ses faiblesses.

Pendant la période de recherche, l'utilisation de ChatGPT est gratuite. Essayez-le maintenant sur chat.openai.com.

Échantillons

Dans l'exemple suivant, ChatGPT est capable de comprendre la référence ("il") au sujet de la question précédente ("le petit théorème de Fermat").

Utilisateur : qu'est-ce que le petit théorème de Fermat ?

ChatGPT : Le petit théorème de Fermat est une proposition mathématique qui affirme que pour tout nombre premier p et tout entier naturel a , il existe un nombre entier x tel que $a^x \equiv a \pmod{p}$. En d'autres termes, si a est un entier naturel et p est un nombre premier, alors $a^{p-1} \equiv 1 \pmod{p}$.

Par exemple, si $p = 7$ (un nombre premier) et $a = 3$ (un entier naturel), alors $3^6 \equiv 1 \pmod{7}$.

Le petit théorème de Fermat a de nombreuses applications en cryptographie et en théorie des nombres. C'est également un élément clé de la démonstration du théorème de Fermat, qui affirme que si n est un entier naturel supérieur à 2, alors n ne peut pas être une somme de deux carrés entiers.

Utilisateur : comment est-il utilisé en cryptographie ?

ChatGPT : Le petit théorème de Fermat est utilisé en cryptographie dans l'algorithme de chiffrement RSA. RSA est un algorithme de chiffrement asymétrique qui utilise des clés publiques et privées pour chiffrer et déchiffrer des messages.

Voici comment RSA fonctionne en utilisant le petit théorème de Fermat :

1. Le destinataire du message choisit deux nombres premiers p et q et calcule $n = pq$.
2. Le destinataire choisit également un entier e tel que e soit premier avec $(p-1)(q-1)$.
3. Le destinataire publie sa clé publique, qui est composée de n et e .
4. L'expéditeur du message souhaitant envoyer un message à ce destinataire chiffre le message en utilisant la clé publique du destinataire.
5. Le destinataire reçoit le message chiffré et utilise sa clé privée (dérivée de p , q et e) pour déchiffrer le message.

Le petit théorème de Fermat est utilisé dans l'étape 2 pour s'assurer que e et $(p-1)(q-1)$ sont premiers entre eux. Cela est important car cela rend difficile pour un attaquant de déterminer la clé privée à partir de la clé publique.

Utilisateur : écris un poème en anglais à ce sujet

ChatGPT : Here is a little poem about Fermat's Little Theorem:

Fermat's Little Theorem, oh so fine
A mathematical concept that is truly divine
It tells us that for any prime number p
And any natural integer a
There exists an integer x
Such that $a^x \equiv a \pmod{p}$



In cryptography, it is very useful
It keeps our secrets safe and beautiful
With its power and logic so true
Our communications remain private, through and through

Fermat's Little Theorem, a valuable friend
In our daily lives, it never ends
It allows us to communicate with ease
And all thanks to its simple equation, oh so neat.

Utilisateur : résumez la conversation jusqu'à présent

ChatGPT : Jusqu'à présent, nous avons discuté du petit théorème de Fermat, qui est une proposition mathématique qui affirme que pour tout nombre premier p et tout entier naturel a , il existe un nombre entier x tel que $a^x \equiv a \pmod{p}$. Nous avons également discuté de son utilisation en cryptographie dans l'algorithme RSA, où il est utilisé pour s'assurer que e et $(p-1)(q-1)$ sont premiers entre eux. Enfin, j'ai écrit un poème sur le sujet en anglais.

Méthode

Nous avons entraîné le modèle à l'aide de l'apprentissage par renforcement par interaction humaine (Reinforcement Learning from Human Feedback, RLHF), en utilisant les mêmes méthodes qu'InstructGPT [InstructGPT est un autre modèle créé il y a quelques mois par OpenAI], mais avec de légères différences dans la configuration de la collecte de données :

(1) Nous avons formé un modèle initial en utilisant un réglage fin supervisé : des entraîneurs humains de l'IA ont fourni des conversations dans lesquelles ils jouaient les deux rôles - l'utilisateur et l'assistant IA. Nous avons donné aux entraîneurs l'accès à des suggestions écrites par le modèle pour les aider à composer leurs réponses. Nous avons mélangé ce nouvel ensemble de données de dialogue avec l'ensemble des données de InstructGPT, que nous avons transformé en format de dialogue.

(2) Pour créer un modèle de récompense pour l'apprentissage par renforcement, nous avons dû collecter des données de comparaison, qui consistaient en deux ou plusieurs réponses du modèle classées par qualité. Pour recueillir ces données, nous avons pris des conversations entre les entraîneurs de l'IA et le chatbot. Nous avons sélectionné au hasard un message, échantillonné plusieurs réponses alternatives et demandé aux entraîneurs de l'IA de les classer.

(3) En utilisant ce modèle de récompense, nous pouvons affiner le modèle en utilisant l'algorithme d'apprentissage par renforcement PPO. Nous avons effectué plusieurs itérations de ce processus.

Etape 1

Collecter des données de démonstration et entraînement supervisé de ChatGPT

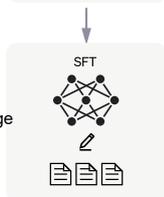
Une question est choisie au hasard dans la base de données



Un entraîneur écrit un exemple de réponse



Ces nouvelles données sont utilisées pour améliorer le modèle GPT-3.5 en apprentissage supervisé



Etape 2

Collecter des données comparées et entraînement d'un modèle de récompense

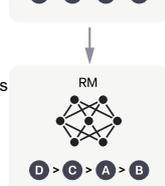
Une question est choisie et plusieurs réponses différentes sont générées



Un entraîneur classe les réponses de la meilleure à la moins bonne



Ces données sont utilisées pour entraîner un modèle de récompense



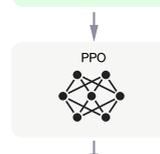
Etape 3

Optimiser le modèle ChatGPT en utilisant l'algorithme d'apprentissage par renforcement PPO et le modèle de récompense

Une nouvelle question est choisie dans la base de données



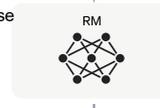
Le modèle qui va être optimisé avec PPO est initialisé avec le résultat de l'entraînement supervisé



Le modèle génère une réponse



Le modèle de récompense calcule une récompense pour cette réponse



Cette récompense est utilisée pour mettre à jour le modèle grâce à l'algorithme PPO



ChatGPT est affiné à partir d'un modèle de la série GPT-3.5, dont l'entraînement s'est terminé début 2022. Vous pouvez en savoir plus sur la série 3.5 ici. ChatGPT et GPT-3.5 ont été entraînés sur une infrastructure de supercalculateur Azure AI.