Bon Pour rappel, on fait encore dans une force me perd pour qu'elle oublié. Dans lequel on veut. Dans lequel on veut entraîner un agent à. Agir du mieux possible pour optimiser sa récompense dans le monde qui l'entoure. Pour ça, on considère un cadre mathématique assez précis et comme je vous l'ai déjà beaucoup répété, c'est différent important de savoir, de connaître bien les les définitions de les des quantités qu'on manipule. Alors du coup, la question que je vous pose c'est c'est quoi ces fameuses quantités et c'est quoi leur définition ? Qui est inspiré ? Alors il y a déjà 3 quantités là qui sont qui sont notées sur le sur le petit schéma, il y en a d'autres dont on a déjà parlé. Ouais, probabilité. C'est c'est quoi alité ? C'est un dans un État donné. Quelles sont les possibilités de changer d'État exactement ? Probabilité qui s'appelle plus exactement probabilité de transition transition, c'est en effet pour un dans le dans le cadre markovien. C'est quoi la probabilité de passer de l'État s à ? Oui. Ouais, qu'est-ce que ça va lui function ? Excellent. C'est la définition que j'avais fonction, c'est très bien. Enfin pas exactement d'ailleurs. Mais cette expérience du quai, dans quel cadre il y a ? Il y a un petit truc qui manque. Ouais. Et elle. Oui exactement c'est l'espérance du gain sachant qu'il se trouve dans l'État SOK la value fonction c'est une fonction comme on dit et c'est pas c'est pas une probabilité, c'est une fonction et après on entrée. Le CE qu'en fait l'État dans lequel on se rend et elle dit à quel point est-ce que l'État dans lequel on se trouve est favorable ou pas dans une partie d'échec ? Si vous êtes à 2 coups d'un échec et MAT, bah votre Valley fonction, elle sera très bonne puisque vous avez une bonne probabilité de gagner la partie. Une autre, une autre idée, policier ouais, politique que policier. C'est un peu comment on va procéder à partir d'un État pour atteindre l'objectif. Disons un peu les règles ou la marque, ça veut dire, ça veut dire ce qu'est ce qu'une politique, c'est ce qu'on essaie de faire, ce qu'on met en place pour atteindre un objectif. Mais mathématiquement, ça veut dire quoi ? On la définit comment la police dans notre cas c'est maximiser la value fonctionne. Par exemple, ouais mais qu'est-ce que c'est fondamentalement la police ? Parce que certes on va essayer de maximiser la value function, mais les policiers la ligne. Définition plus générale que ça. Par exemple, on a parlé de Random policy. Qui est intuitivement c'est juste, on fait n'importe quoi et on regarde ce qui se passe et exactement c'est une stratégie. Mais du coup la la définition exacte c'est quoi ? Ouais. En fonction de quoi, ce qu'il prend en entrée pour Lucie. Donc pour Lucie c'est la probabilité de faire l'action a sachant que je suis dans l'état s donc je suis dans ma partie d'échec et quelle est la probabilité que je fasse le fameux coup d'échec ou pas, par exemple la grédy policier ? La politique glouton. Et bah elle va prendre systématiquement le meilleur coût qu'elle sait faire. Donc si on connaît la value function, on fait le meilleur coup, on fait le coût qui va nous mettre dans la meilleure des situations et ça fait qu'on a normalement une politique qui est censée être optimale. D'autres choses ? Modèle modèle, ce que c'est un modèle. C'est très large comme vous. Je. On va voir. J'ai entendu environnement aussi. Ouais, ce que c'est environnement. À partir des actions, il peut donner l'État à son bien. Ouais, c'est ça l'environnement, c'est le ça peut être un modèle typiquement, mais c'est ce avec quoi on interagit, c'est-à-dire l'environnement, on ne va rien optimiser, c'est ce qu'on nous donne. Donc l'environnement ça va être la partie d'échecs et l'environnement, il va nous dire, t'as pas le droit de déplacer ton cavalier de de de de pas en avant comme ça. C'est c'est pas comme ça que fonctionne l'environnement. L'environnement il va dans le cadre d'un d'un robot, il va dire bah le robot il veut accélérer, il veut-il veut aller plus vite dans l'environnement, va lui dire Ah bah très bien, ça va appuyer sur l'accélérateur. Bah maintenant ça veut dire que ta vitesse va augmenter de 1 km. Donc on le contrôle pas. Nous on peut que essayer de faire des actions tout comme mon action maintenant c'est essayer de vous apprendre le enforcement Learning. L'environnement il va dire Ah en fait lui lui au fond il a rien compris mais c'est pas c'est pas moi qui le contrôle. Et donc tout ce que je peux c'est optimiser la manière dont je vous parle et la manière dont dont je je vous présente des choses pour espérer que vous comprenez le mieux possible. Okay ? Oublié quelques petites choses je pense. Ah oui, l'agent. Un agent donc, c'est un peu la le premier. Le premier, comment dire le premier élément, l'environnement, ça a été dit. L'action, ça, on l'a dit aussi, mais l'action, c'est directement liée à la politique. L'État, donc. Sa définition exacte, c'est ça, c'est l'ensemble des informations qui déterminent le l'État dans lequel on se. Donc ça va dépendre de ta dépend des environnements, et cetera. Et la récompense ça on ça vous l'avez pas dit ? Mais c'est le le moyen qu'a l'environnement pour dire à l'agent que ce qui fait, c'est bien ou pas. Donc ça va être les notes que vous allez avoir à la fin du cours. Ça va être à quel point vous allez être heureux dans votre vie en fonction, ça va être si vous gagnez votre partie d'échecs, et cetera. Ok. Maintenant qu'on a, on a fait un petit point dessus. Je me permettre aussi, je pense, dans le doute, dans le doute, autant de reprendre. Je vous remets. Stop. Cette petite. La petite fiche que je vous ai envoyée ? Où on a pareil les définitions un peu plus mathématiques, donc là de policiers, et cetera. La fameuse définition de fonction, tout ça c'est des choses que vraiment faut connaître par cœur. C'est pas très compliqué hein ? Le gain c'est quoi ? C'est la somme des récompenses. Sauf que on veut prioriser les récompenses à court terme mais quand même pendant qu'on A long terme, donc on va pondérer par un certain gamma, et cetera. La la value function comme votre camarade vous l'a dit, c'est bien l'espérance du gain sachant qu'on est à l'État. Il y a l'histoire de l'équation de Bellman qui est un rappel. Elle sert à quoi les passions de bellman ? Ouais, à partir de l'État précédent, savoir tous les États, je peux le dire un peu plus fort, à partir de l'État précédent, on est censé pouvoir savoir connaître tous les États passés par exactement. Là tu vois dans l'équation de Bellman ça a 2 termes qui qui est qui interviennent, t'as s un et s 0. Donc c'est pas tous les États précédents. Par contre en effet, il y a une histoire de. Il y a une histoire d'État précédent. Et ça, ça dit que si tu connais l'état à un, tu peux rétro propager sa valeur sur l'état s 0 si tu connais la récompense entre. État actuel, il faut reconnaître précédent, c'est ça un des États précédents. Du coup la value section exactement après le coup c'est pas forcément tous les États parce que comme on va le voir aujourd'hui on c'est pas évident que tu puisses propager à partout. Nous jusqu'ici on l'a fait sur des très simples, on a, on a travaillé sur des modèles de labyrinthe. Hop, on travaille sur des modèles de labyrinthe dans lesquels on pouvait en effet se permettre de se permettre de tout, de tout calculer. Mais là, on va voir. Aujourd'hui, on va commencer à faire des choses plus compliquées. Parce que dans la vraie vie on peut pas explorer toute labyrinthe. Oui d'accord, dans la vraie vie vous pourrez jamais tester tous les États possibles et rétro propager avec la les cousins de Bellman. Donc jusqu'ici, on a fait ce qu'on appelle du timing programming. Ou la programmation dynamique, c'est quoi ? C'est en fait je je sépare mon très très gros problème en plein de petits problèmes et de les résoudre tous, tous, tous, tous, tous. Et du coup à la fin je récupère ma valeur sur sur toutes mes sur sur toute ma grille. Sauf qu'aujourd'hui on va commencer à faire des à trouver des algorithmes qui nous permettent de quand même avoir des approximations. Quand on peut pas se permettre de d'explorer toute la grille donc on va commencer notamment à parler de d'épisodes donc je sais pas si si quelqu'un se rappelle j'en ai rapidement parlé les dernières fois c'est qu'un épisode. Ouais, exactement. Un chemin qu'on fait, c'est à dire dans le labyrinthe. Je laisse, je laisse mon agent dedans, je regarde ce qu'il fait, j'attends la fin du labyrinthe et là j'ai une trajectoire. Trajectoire, épisode c'est un peu des des termes qu'on va pouvoir un peu utiliser manière interchangeable. L'épisode ça va plutôt être l'ensemble de l'environnement, agents, actions, donc ça c'est tout. L'épisode c'est ce que vous allez regarder en vidéo à la fin sur sur collab quoi. Et la trajectoire, c'est plutôt l'ensemble des étapes que va traverser le le modèle, la trajectoire, ça va être la ligne que va tracer le l'agent dans le. Dans le dans l'environnement. Ok, donc votre petit rappel de notion dont on a parlé déjà parce que on. En une force majeure on a plein plein d'algorithmes qui sont qui ont été été pas un peu faire une une taxonomie de tous ces algorithmes pour s'en rappeler parce qu'on va se dire Ah tiens mais y a y a salsa, y a tu learning YAPOTRPOYA des trucs, vous allez voir, il y a plein de plein d'acronymes dans tous les sens. Et donc pour s'aider et pour mieux comprendre ce que fait chacun de ces algorithmes. Ou on a une taxonomie, c'est-à-dire on a une manière de les décrire. Donc on a déjà parlé la semaine dernière de modèle Free modèle base. Normalement je sais pas si ça ça doit encore vous parler. C'est pas grave si vous en connaissez pas la définition en encore un moment. Donc y a ça y a offre aussi offre aussi y a value base ou ou action base policy base. Donc par exemple dans le cas de la de la progression dynamique, on a vu que on avait un algorithme où on trouvait toute la value function avec la programmation dynamique. Et à partir de cette value function, là on en a, on en déduisait la politique optimale. Parce que si on connaît la value function pour tous les États, alors c'est évident de trouver la politique optimale. On trouve juste, on maximise juste notre récompense à l'état d'après. Donc ça ça s'appelle un algorithme qui est. Value base et qu'est ce que m'expliquer pourquoi ? Exactement. On a modélisé absolument pas la politique. Quand on trouve la solution on s'en fout, juste on explore tout et on se dit Ah bah là c'est une position qui est bien là c'est une position qui est bien, et cetera. Et donc à aucun moment du de notre algorithme on ne modélise la politique. Simplement on sait que à la fin on aura la politique optimale parce qu'on a trouvé la value function. OK et donc voilà, il y a plein de termes comme ça, value base pour wikibase et cetera qui sont. Qui sont en réalité c'est une histoire de taxonomistes mais ça vous permet quand vous allez vous rappeler vous allez vous trouver devant un problème de fortune turning vous allez vous dire Bah en fait là je sais pas, j'ai pas possibilité de modéliser ma poule ici donc je vais forcément devoir faire du value base parce que je peux pas je peux pas faire si je peux pas faire ça. Et là ça va vous aider à vous guider vers l'algorithme qui va. Qui aura du sens et dans le dans l'autre sens aussi. Si vous dites Oh \*\*\*\*\* pipiou enfin le mec dans ce papier, là où on fait de la recherche ou même quand vous réimplémenté quelque chose Open, I expect you pour faire du rls. Ben en fait \*\*\*\* o Ah mais je me rappelle que pipio c'est de l'acteur critique mais offre politique parce que ils utilisent tel ou tel truc. Et donc vous aurez même pas besoin d'aller revoir le le code, ça vous permet de vous vous rappelez de comprendre un peu ce qui se passe à l'intérieur. Donc bref, tout ça pour vous dire que là on va, on va commencer à à mettre ces termes là sur les algorithmes, on va poser parce qu'on va voir aujourd'hui nos premiers algorithmes de RL vraiment utilisables dans la vraie vie. Et donc on va au fur et à mesure. Vous allez voir que j'ai essayé de les surligner à chaque fois pour que vous fassiez bien la différence. Mais on on va voir à chaque fois comment est-ce qu'on peut différencier tel ou tel algorithme par rapport à ce qu'on a vu précédemment ? Et donc première notion qui est un peu. Qui est un peu sous-jacente à tous les algos qu'on voit, c'est le l'exploration versus exploitation. Avec toujours ce cet équilibre entre si je veux connaître tous les États avantageux de ma. Si je veux connaître tous les États avantageux dans mon système, bah il faut les former un minimum. Sauf que quand on explore, Ben on se fait mal parce que on teste des choses qu'on n'a jamais faites et potentiellement ces choses là elles sont pas du bien. Heureusement que dans la vraie vie on a des parents qui vous disent il faut pas mettre les doigts dans les prises mais dans l'idée votre agent il y aura pas des parents pour lui dire que c'est pas bien. Donc dans l'idée il va tester, il va s'électrocuter. Il va se dire bon bah prochaine fois je le ferai pas. Voilà et l'exploitation, c'est une fois qu'on a un peu de connaissances sur le. On a un peu de connaissances sur le milieu et bah on on exploite ce qu'on connaît déjà, ça dure qu'on va aller, on va aller vers les les actions que pour lesquelles on sait qu'on aura des bonnes récompenses. Voilà. Et donc vous allez voir dans les algos qu'on va voir, y a toujours un peu cette idée de à quel point ce que pendant l'apprentissage je vais faire de l'exploration pour être sûr de bien tout comprendre. Et à la fin de l'apprentissage je vais plutôt faire de l'exploitation pour avoir la politique qui soit la meilleure. Ok c'est clair pour tout le monde ça. Donc il y a toujours y a toujours un peu ce cette idée-là. Donc par exemple voilà le robot dans un labyrinthe. Bah si jamais il a pas encore trouvé le chemin optimal. Bah ça vaut encore le coup d'aller tester des endroits un peu sombres qu'il a pas vu du labyrinthe pour pour pour tester si jamais en fait un raccourci ou j'en sais rien. Un système de recommandation c'est pareil, on en a déjà parlé. Si vous savez que votre utilisateur il adore les vidéos de chat sur Youtube, vous êtes sûr que vous aurez une récompense très facile, c'est vous le mettez que des vidéos de chats. En fait, ça se trouve il va aimer les vidéos de chiens où ça se trouve en fait un truc complètement différent. Il va aimer un autre sujet qui a à priori aucun rapport avec les vidéos tcha. Et donc en fait ça vaut le coup de caser parfois une petite vidéo de complètement autre chose et de voir si l'utilisateur clique dessus. Et ça se trouve allez trouver qu'en fait il adore encore plus les vidéos de chameau. Voilà donc là il faut-il faut tester et vous allez voir vraiment dans vos objets de recommandation, que ce soit sur Youtube, sur sur tik Tok, sur Amazon, vous allez voir parfois Ah \*\*\*\*\*\* cette vidéo elle est pas trop en rapport à ce que j'ai d'habitude. Mais c'est l'algorithme de recommandation qui essaye des choses pour essayer de mieux vous cerner. Alors ouais, y a une question. De recommandation comme y a du nouveau contenu qui arrive tout le temps permanent. Est ce que ça vaut pas le coup de temps en temps justement de réexplorer à nouveau pour pouvoir mettre à jour justement l'exploitation ? C'est ce que je dis. Et quand t'as du nouveau contenu, t'arrives, tu vas le catégoriser. Et donc si ton utilisateur tu sais qui suit telle ou telle chaîne, bah là t'auras des systèmes qui qui qui permettent un peu de de de mettre un score supérieur à des vidéos qui sont récentes. Tu vois si tu sais que tu suis tel ou tel producteur, Eh Ben il va voir. Ah bah y a une vidéo de tel producteur. D'habitude mon client il regarde tout donc forcément je vais la mettre en haut de la pile parce que c'est probable qu'il va cliquer dessus. Il y a ça et là c'était plutôt ce que ce que j'expliquais, c'était plutôt une vidéo qui est pas du tout en rapport. Parce que dans ce cas c'est un peu facile tu vois, si si tu devais choisir les vidéos que tu mettais, tu mettais ton utilisateur, tu le ferais. Tu sais que il il suit tel ou tel youtubeur ? Ben en fait si je vais lui mettre comme ça, je suis à peu près sûr qu'il va cliquer. Donc là t'as déjà de l'information là je parle plutôt de quand Youtube il vous met une vidéo complètement Random car rien à voir avec ce que vous Regardez d'habitude. Juste pour tester si peut-être en fait ce peut être ça vous plaît ? C'est ahh non c'est un algorithme parce que moi j'ai jamais une vidéo. Bah c'est un peu éloigné, mais pas. Il se base pas sur quelqu'un d'autre, par exemple, si, si, si, si, mais tu vois. Euh. Euh c'est un c'est un entre-deux. Si tu veux t'as pas t'as le choix entre vraiment je vais mettre que ce qu'il a déjà regardé versus des trucs complètement Random. Peut-être qu'il mettra pas des trucs vraiment complètement Random. Mais peut-être qu'il va se dire Ah mais il mon utilisateur, il est proche de tel ou tel autre utilisateur qui regarde ça, donc peut-être qu'il va aimer, mais c'est pas évident tu vois. Donc là c'est déjà un peu plus de l'exploration au sens où il te connaît pas vraiment, mais il va essayer des choses, OK ? Pardon, pas encore commander system, mais voilà ce que vous voulez donner des générales. Et bon, voilà le fameux bénéfice. Donc exploitation pour des agents de trading. Pareil en trading vous faites pas d'exploration en fait vous entraînez un offline sur vos données, vous avez votre algo ? Il prédit bien, il prédit le marché et vous faites de l'argent dessus. Mais vous avez pas envie d'explorer, d'explorer quand vous faites du trading pareil quand quand votre IA. Vous l'avez entraîné et vous faites un un championnat. Enfin en fait un un concours contre un un joueur humain pour vérifier que l'i a est meilleur que l'humain. Bah vous allez pas faire de l'exploration là, vous avez juste exploité ce que vous avez déjà appris en dans votre entraînement. On capitalise sur tout ce qu'on connaît déjà. Ok, très bien. Maintenant on va passer sur vos premiers algos vraiment de RL. J'avais déjà utilisable dans la vraie vie et on commence avec. Enfin et globalement dans le cours, ce sera beaucoup des des des algorithmes qu'on va catégoriser comme ça. On appelle les méthodes modèle Free et est-ce que quelqu'un se rappelle un peu modèle free ? On en avait parlé rapidement. Où est-ce que ça parle à personne ? Non non, mais très bien, c'est pas grave. Donc. Une méthode modèle Free, c'est quoi ? Une méthode maîtrise, ça veut dire que on ne va pas modéliser l'environnement dans lequel on se trouve. Ok, par exemple dans notre labyrinthe. Ce qu'on faisait c'est que on on cherchait la value function pour toutes les positions, mais on s'est jamais demandé où est-ce qu'on se trouvait. En fait, on avait juste un État qui était une coordonnée dans un labyrinthe. Et on se disait, est-ce que c'est bien resté dans cet état là ? Est-ce que je devrais aller à la case d'à côté ? Mais jamais. On s'est dit en fait là je suis dans un labyrinthe et si je vais à droite, Bah je vais me rapprocher un peu de la sortie ou j'en sais rien. C'est un peu plus dur pour la viande, mais par exemple. Dans le cas. Dans le cas d'un de jeux à tarif ou de de jeux de jeux vidéo. Si vous modélisez pas votre environnement, ça veut dire que vous Regardez juste votre écran de pixel, vous dire OK, ça c'est mon état et directement sans se poser aucune question. C'est quoi l'action que je dois prendre ? En réalité, vous êtes beaucoup plus intelligent que ça quand vous, quand vous jouez un jeu vidéo, quand j'ai j'en sais rien, un 1GTA par exemple, ce qui ressemble un vrai avis. Donc c'est pratique. Vous êtes en train de conduire une voiture et Ben en fait vous allez prédire ce qui va se passer. Vous savez que Ah \*\*\*\*\* y a une voiture qui arrive en face, je vais ralentir. Je vais tourner. Et Ben ça, ça veut dire que vous modélisez l'environnement, ça veut dire que vous faites une prédiction sur ce qui va se passer. Vous vous dites, Ah mais si je fais telle action, ça veut dire que probablement dans 4 ou 5 secondes, quand le joueur d'à côté il aura fait son truc, Ben je vais prendre ça dans la gueule donc c'est pas une bonne idée. Donc je vais faire une autre action justement pour éviter la situation que je vois arriver. Et ça, ça s'appelle modéliser l'environnement. Et ça ça coûte super cher parce que c'est super dur de de prédire un environnement. En fait de la prédiction d'un d'un écran de pixel, c'est c'est très difficile même. Enfin même les les l'algo les plus les plus avancés aujourd'hui j'arrive pas de manière robuste. Donc les premières méthodes qu'on a en RL, c'est des modèles, des méthodes qu'on dit modèle Free, c'est à dire que on s'en fout de prédire ce qui va se passer, l'environnement. Simplement on sait de l'État dans lequel on se trouve et on essaie de faire au mieux en fonction des connaissances qu'on a à l'instant T on va pas chercher à prédire l'instant t plus OK. C'est ça modèle, on n'a pas de modèle. Et lucide pour l'environnement. Euh TAC TAC TAC. Ouais ouais, caractéristique. Donc ce qu'on retrouve beaucoup dans les dans les méthodes modèles Free, c'est le fait qu'on va évaluer une certaine politique. Ou une certaine value function ? Ça dépend si on est policy base de value base OK. La notion de contrôle encore une fois si vous êtes pas familier avec le contrôle c'est pas. C'est pas forcément d'utiliser ce terme là mais le renforcement learning et le contrôle c'est très proche donc on est on est obligé de faire de faire cellard mais à c'est l'idée que vous vous pouvez apprendre sur le tas pour trouver le meilleur contrôleur donc contrôle pour. Faire avancer votre argent ? Euh, et même si de base ça paraît assez. De de ne pas modéliser l'environnement. On va se rendre compte que des modèles Free peuvent être applicables sur des environnements très compliqués et donc même sans aucune notion de de prédire l'environnement, on va voir qu'on va. On peut s'en sortir très bien. Donc voilà, on a beaucoup d'applications un peu partout parce que vraiment à tous les domaines et. Dans ce dans ce genre d'acheter. Ouais Bah est ce que dans le labyrinthe on connait environnement ? Mais on en faisait une reco propagation du. Ça n'empêche pas simplement d'en. Dans le labyrinthe là avec je sais pas si on si on fait le le premier algorithme qu'on a vu c'était policy itération où on a une politique et on on trouve la fonction associée et on itère là-dessus. Et. Dans ce cas-là, à aucun moment on a mené l'environnement. À chaque fois, tout ce qu'on fait, c'est se dire je veux la vallée fonction, la value fonction c'est quoi ? C'est une fonction qui prend d'entrer l'État dans lequel je suis. Qui me qui me prédit à quel point c'est bien ? À quel point le gain dans à partir de cet État là va être bon ou pas ? Mais à aucun moment ça veut dire que je prédis ce qui va se passer dans l'environnement. C'est pas la même chose ce que je fais. Ce que je prédis, c'est uniquement le gain. Je prédis pas ce qui va se passer. C'est comme si quand tu joues à un jeu de voiture, je prédisais mon rang à la fin de donc c'est à dire ma récompense. Mais à aucun moment je prédis que la voiture d'à côté va me rentrer dedans. Ce que je veux dire ? Et c'est très bien. Tu peux cette question parce que c'est important en effet, on prédit quand même des choses, on prédit le gain mais on a modélisé pas l'environnement. Et c'est pour ça que ici y a marqué pas de modèle explicite pour l'environnement. Parce que en pratique, typiquement dans le labyrinthe, quand on a une vallée function on a quasiment accès au labyrinthe et que on on voit qu'on peut reconstruire le labyrinthe à partir de la de la fiction, on voit en fait à chaque fois pour une récompense de moins un pour chaque. Pour chaque time step, on avait remarqué que la fonction elle était 0. Sur le l'arrivée puis puis dans chaque case, c'était moins la distance à l'arrivée, vous rappelez ? Bah là dans ce cas-là on peut reconstruire la grasse. Donc c'est pour ça qu'on dit que c'est pas fait de manière explicite. Ou là c'est comme un très simpliste mais. Mais fondamentalement, dans les équations qu'on a marquées, jamais on a, on a, on a écrit eux de quelque chose, environnement on n'a jamais fait de prédiction de c'était +1. On a simplement dit, Je sais que j'ai un état arrêté, j'aurais un État +1 je le connais pas tout simplement. J'essaie d'évaluer le gain que j'aurais à cet endroit là, parce que quand j'ai fait une étape, une fois que je l'ai fait. Donc j'ai j'ai pas modélisé l'environnement, juste j'ai fait l'étape et j'ai reçu une récompense. Et à partir de ça je rétro page j'ai Ah bah là j'ai une bonne récompense. Donc l'état d'avant il était bien. Mais c'est une question, c'est donc. Toc. Alors les méthodes qu'on va voir aujourd'hui, donc là c'est un peu quand vous allez relire les Slides, c'est ce sera pratique pour vous rappeler tout ce qu'on a vu. Ou globalement, on. On va avoir 2 types de méthodes aujourd'hui. Des méthodes de monte-carlo ? Je sais pas si Monte-Carlo ça parle à quelqu'un ici. J'avais fait des ouais très bien, on aura un rapide point dessus. Mon méthode de Monte-Carlo, c'est plutôt des méthodes pour vous donner une première intuition où on va faire un épisode en entier. Donc on aura une estimation qui est non biaisée du gain qu'on va récupérer. Parce qu'on l'a, on, c'est une vraie valeur du gain, c'est pas une valeur qu'on aura prédite. Il y A partir de ça, on va essayer de de modéliser de manière statistique les récompenses au cours du temps. Et ensuite ? Donc ça c'est Monte-Carlo. Donc là le le la clé c'est ça c'est on apprend à partir de l'épisode complet et. Et du coup on a, on a des des, des symboles qui sont moins biaisés que dans le 2e cas où. Le. Ce qui s'appelle des des modèles de Temporal de France où cette fois on se base un peu plus sur belman. Ce qu'on a vu la dernière fois c'est à partir de ce n'est +1 je peux déjà être propagé sur RT même si j'ai pas terminé mon épisode. Donc le temporal différence c'est un peu ça, c'est je fais une étape, je j'observe une récompense, je propage derrière moi même si je connais pas encore la fin et je continue. Hop je reprends pas, je reprends pas J et cetera, ce qui fait que je peux être biaisé parce que si jamais je termine sur par par grande chance je termine spoil sur mon arrivée. Ben je vais me dire en fait ça se trouve tous mes États. Dans lequel je suis passé, ils étaient super. Alors que ça se trouve je savais pas mais au milieu du chemin j'étais à une case de mourir parce que je suis passé juste à côté d'un gouffre ou j'en sais rien. Et dans ce cas-là ma value function elle est biaisée vers le positif parce que en fait je suis tombé, enfin je suis pas tombé alors que j'aurais peut-être dû et dans et à l'inverse. À l'inverse, si je n'arrive pas à terminer le jeu, bah toutes les évolutions que j'aurais observées elles vont être elles vont avoir une valeur nulle, enfin une mauvaise valeur. Alors qu'en fait ça se trouve, j'étais une case de gagné. Bref, on on va le revoir, donc dans l'idée c'est. D'un côté Monte Carlo c'est plutôt des épisodes complets et on on fait un peu le débriefing après avoir terminé l'épisode les temples différence c'est online, c'est-à-dire ? Pendant l'épisode je peux regarder derrière moi et me dire Ah bah j'aurais trop propage tout de suite et ça me fait plus d'informations. Donc en l'occurrence on va observer que les les méthodes de partum redevance convergent plus vite parce que on fait plus de on fait plus d updates rapidement je peux faire une update par par step de mon environnement là où pour monter Carlo je vais faire une très grosse update mais à la fin de mon épisode donc ça coûte plus cher. Donc voilà, c'est résumé ici. Ça, ce sera vraiment le la slide de résumé. Ensuite ? Les fameuses notions sur l'unique qu'on va voir aujourd'hui et qu'il faudra bien avoir en tête. Modèle Free on l'a vu, Monte Carlo on l'a vu, temporal différence on l'a vu et bootstrapping on l'a pas encore vu. Mais vous pouvez déjà le lier dans votre tête avec temporisation. Parce que le bootstrapping en gros, ça veut dire on va recoller par-dessus quoi. C'est ça bout de strap, c'est bootstrap, c'est je prends, je prends mon câble, hop je branche un autre, je les mets ensemble et j'allonge mon mon temps comme ça c'est du bootstrapping et du coup quand on fait de la Template defence, souvent c'est ça c'est j'ai mon épisode, j'ai fait une étape de plus et je recolle ce que la nouvelle information que j'ai avec tout le reste. Donc ça s'appelle faire du Bootstrapping oui. Dans le sens où si l'environnement devient trop grand, en fait c'est pas applicable tout à fait. Tu vas voir ce genre de ce genre de choses et c'est pour ça. Pardon on parlait plus en détail hein, c'est c'est complètement. T'as t'as bien l'intuition de base que Monte-Carlo c'est moins biaisé parce que tu t'attends d'avoir une une vue globale avant de d'optimiser ton truc. Par contre, ça coûte cher parce que tu dois faire un épisode, mais il y a des, il y a des. Des cas en RL ou. Où en fait ton épisode est très court et donc en fait tu peux te permettre de faire du Monte Carlo parce que. Parce que tu peux hein. Et c'est pour ça que toutes ces notions là c'est important parce que chaque problème de R va être différent et vous allez appliquer en fonction de du problème sur le enfin sur lequel vous êtes-vous allez appliquer le Monte Carlo ou pas, faut connaître. Ah je fais du Monte-Carlo donc je suis pas biaisé mais ça va être plus lent à converger. Mais peut-être que vous voulez un truc ultra stable tout de suite, donc vous préférez Monte-Carlo. Être vous voulez un truc ? Qu'on Darty quoi, un truc rapide ? Bah là je vais plutôt faire du temptations parce que je sais que ça marche plus vite, mais attention c'est instable. Donc ça vous allez jamais convergé avec du temple du vent alors qu'avec du Monte Carlo Ça va plus lentement. Mais en fait vous convergez donc finalement vous allez plus vite puisque j'aime pas du tout, c'est juste qu'on la jouera pas. Voilà, c'est un peu toutes ces idées, faut, faut pas, y a pas de. Été dans ce cours là, franchement il y a pas de meilleur modèle, et cetera. Vous allez voir, on va construire des on va construire des des algorithmes petit à petit l'un après l'autre. Mais faut pas se dire à lui, il est forcément mieux que le précédent, c'est juste, il est différent, il y a, il y a des nouveaux éléments dessus. On va être parfois meilleur sur tel ou tel problème, mais mais avant tout il faut toujours garder l'esprit tel rythme, il a tel avantage mais aussi tel défaut et donc faudra l'utiliser ou pas dans certains. Donc visiblement, vous connaissez déjà. Je me suis même permis de faire un petit rappel sur le Monte Carlo. Je Monte Carlo, c'est un c'est un paradis qui est beaucoup plus large que le renforcement learning. Donc là c'est un super gif qui est sur Wikipédia. Si vous voulez estimer la valeur de PI, et Ben vous pouvez le faire comme ça. Parce que là l'air de mon l'air de mon quart de cercle, c'est exactement PI. Et donc si je fais un sampling. De plein de points. Dans un carré de un par un et que je mesure si la différence, enfin si la distance, donc X carré plus y carré est inférieure à un. Alors en faisant ce rapport là je vais avoir une estimation depuis donc ça c'est typiquement une méthode de Monte Carlo où ? Je je fais du sampling aléatoire sur sur ce que j'ai et j'obtiens une. Et j'obtiens un estimateur de d'une variable que je. Là c'est un cas très très simple, mais le la le pas. Enfin dire la la théorie de Monte-Carlo ça apporte. Y a plein de thèmes très beaux sur bah si je sens, enfin si je fais mon camping de telle ou telle manière, je vais converger avec telle rapidité et on j'aurai tel ou tel résultat. Et et donc globalement, ce que vous pouvez retenir c'est on est, on est, on pardon ? On estime des quantités complexes par des moyens statistiques. Ok. Euh qu'est-ce que je voulais dire de plus là-dessus ? Non c'est bien, donc on m'a appliqué Monte-Carlo, c'est très connu donc faut faire attention toujours ces problèmes là quand on fait de l'i a du enfin du RL, on croit que on cette notion nous appartient la, pas du tout. Monte-Carlo ça devient de dans ça vient d'un milieu complètement différent. Donc Monte Carlo, on va voir qu'on va pouvoir faire une méthode modèle Free pour estimer des value function parce que si je peux estimer PI avec des Random Samples dans dans un carré, je peux bien estimer une value function en faisant des samples dans mon journal non ? Donc vous allez voir, on va, on va trouver un algorithme qui fait exactement ça. Et donc l'idée c'est qu'on fait un sampling sur un épisode en entier, puisque pour avoir la value function il faut le gain. Donc c'est l'espérance du gain. Et le gain c'est bien une notion qui est attachée à une trajectoire en entier, pas juste un État. Donc on va faire une espérance d'un gain et donc on va estimer le gain et donc pour estimer le gain. Il faut faire des épisodes. Tac TAC, et donc on va pouvoir améliorer nos politiques en fonction de des gains qu'on a récupérés. Gains return c'est pareil, c'est encore désolé mais c'est interchangeable. Donc return ici c'est bien un grand g qui est la somme des de la comparaison. C'est clair tout le monde ça. Ok. Alors ? On veut faire de la prédiction, on veut estimer notre value function vpi. En fonction de notre. Avec du Monte-Carlo donc. On fait un épisode entier, OK, on a accès, on est à un l'action, un que j'obtiens avec PI ma récompense, et cetera. Donc là j'ai une ça, c'est une trajectoire. Je calcule mon gain. Ok et ma value function d c'est facile c'est l'espérance de mon gars. Donc là super facile. Donc là on fait, ça s'appelle du du policier. Évaluation c'est on a une politique, on fait plein de trajectoires et on estime la valeur de cette politique en estimant le gain, le gain qu'on a à la fin, c'est clair ça, pardon, le truc le plus basique qu'on puisse faire. Ouais. Ouais, ouais, c'est. Une récompense mais du coup tu passes dans le. Franchement, j'aurais pu mettre un aussi, c'est pas. Et du coup c'est comme si t'avais ta récompense d'avant enfin bah c'est pas faut pas faut pas. Ouais, exactement, ça c'est une trajectoire associée à un épisode. Exactement, on calcule la loi des grands nombres ne dit que. Si je fais la moyenne de tous les gains que j'obtiens d'abord, alors à la fin j'aurais une valeur à peu près satisfaisante. Et vous imaginez que probablement que il me faudra être beaucoup, beaucoup, beaucoup, beaucoup de run pour que ça marche. Mais en théorie, ça marche. Donc si vous avez un environnement, vous pouvez faire plein d'épisodes autant que vous voulez. Bah là c'est c'est bourrin, mais ça marche tout de suite. Non c'est justement et c'est pour ça que c'est pas biaisé. Tu dépends pas de l'épisode que t'as fait avant, tu fais vraiment un point dans ton espace un point 2 3, et cetera. Ouais. Tout l'environnement oui, mais en fait. Ouais, ce que je dire c'est qu'on ne modélise pas l'environnement, c'est-à-dire que là je fais une trajectoire, tu vois ? Mais mais je cherche pas à la prédire cette trajectoire, simplement je suis dans mon état s un je je fais une observation, je fais une une une action, j'observe une récompense et à aucun moment je me dis Ah mais l'environnement ça se trouve il va me faire telle ou telle crasse. Ok donc là je modélise pas, mais je fais quand même trajectoire dedans et et Ben t'as dit un truc. Faire attention à dire on explore tout l'environnement, ça c'est pas vrai, je fais juste une trajectoire. Si par exemple, j'évalue la policy aller tout droit. Bah mon environnement, si je si mon.de départ c'est toujours ici, Ben je vais juste faire ça. Hop à la fin je vais me prendre la porte là hop je vais me prendre une récompense négative, je me dire C'est pas bien et si je change pas ma politique et que tout le temps je fais cette trajectoire là bah je vais pas, je vais pas explorer tout mon environnement tu vois. C'est pour ça que typiquement dans Monte Carlo. Vu qu'on fait un un, une estimation de. De comment dire à partir de de plein de trajectoires. Là y a une notion d'exploration qui est très importante, parce que si j'ai une politique qui est trop déterministe au début, je vais jamais explorer tout mon environnement. Et donc typiquement dans les méthodes de Monte-Carlo on va faire des politiques du type Random où justement on sait que sa chambre à faire plein d'épisodes alors on va aller un peu partout, on va laisser l'agent explorer quoi et. Et on espère du coup trouver des des meilleures, des meilleures valeurs. Ok Ouais. L'environnement. Mais on a quand même besoin de savoir quand est-ce qu'on se prend une rivière de temps. Oui, mais ça, ça, c'est l'environnement qui nous donne ça. C'est la base du renforcement learning. Si t'as pas de récompense, oui tu peux rien apprendre. Comme le dit Environnement, de savoir quand est ce qu'on. C'est pardon, c'est quand c'est pas comme le fait d'avoir une récompense ou pas un état de données. Ouais c'est si on le définit, c'est pas modéliser l'environnement ça parce que ça dépend quand même de notre environnement. Bah c'est comment dire, c'est le cadre du R en fait. Ouais, juste on est dans un cas où t'as un agent qui fait des actions dans un environnement, l'environnement il le connaît, pas simplement l'environnement. Il lui dit Bonjour t'as le droit à telle action à tel moment et merci t'as pris telle action, je te donne telle récompense, c'est bien, c'est pas bien. C'est tout donc c'est une boîte noire. Tu mets genre en entrée et ton action en sortie, tu te prends une baffe ou tu prends une récompense ou pas ? Quand tu le programmes oui mais en pratique vous là dans ce cas-là vous aurez jamais à programmer un événement par exemple dans le cas de je suis roboticien et je veux apprendre un robot à faire un salto parce que je m'appelle Boston Dynamics et Ben oui je vais devoir modéliser mon environnement. Je vais devoir modéliser mon mon Atlas là. Et je vais lui devoir faire un modèle physique à l'intérieur. Qui va le dire ? OK quand le modèle il bouge le bras, bah il va se passer ça dans l'environnement par contre. Donc ça c'est moi ingénieur qui va créer cet environnement là. Par contre mon agent quand je vais faire du R après je vais dire OK, ça c'est la trajectoire optimale d'un salto apprend à partir de l'environnement. Le modèle il va dire OK. Je je veux-je j'active le mondo et cetera. Il va jamais se dire en fait si j'active mon dos maintenant, c'est à dire Je vais tomber dans 3 étapes quoi. Simplement il optimise son truc et c'est et l'environnement est à nouveau une boîte noire pour lui. Ok l'environnement est nécessaire bien entendu si vous voulez faire du RL en pratique vous votre problème c'est pas de miser environnement, c'est l'ingénieur d'à côté qui va le faire. Parce que vous allez dire OK, moi je suis-je suis un. En, comment dire ? 1I Scientist, je peux vous, je peux faire du R sur ton robot. Moi j'ai besoin d'un environnement qui me donne des récompenses ou pas. C'est juste pour imaginer un peu ça avec faire un parallèle avec la vie. On est né dans le monde qui le monde, on a pas eu à le modéliser en solde il est-il est connu, il est comme il est mais en fonction de en fait c'est toi quand tu né t'as aucune connaissance du monde avec lequel tu vas interagir mais le monde te file des récompenses ou pas. Très bon exemple. Je euh. Oui donc en soit on connaît quand même le débat ou pas. Oui tu as accès à ton état et c'est la fameuse observation dont je vous parlais au début. Parce que toi être humain tu observes ton environnement. J'ai pas accès à toute l'information en même temps. Je sais pas quelle est la couleur de ton caleçon que t'as mis aujourd'hui, mais je vois, je vois la classe et donc j'ai accès à j'ai fait une observation sur l'environnement avec mes yeux, avec mes oreilles, avec mes mains, et cetera. Ça c'est toute l'information que j'ai qui fait que je suis dans un État donné. Et c'est vrai qu'il faut garder à l'esprit que dans beaucoup d'environnements de RL, l'État au total, c'est-à-dire la position des atomes dans l'univers à l'instant T maintenant je le connais pas. Par contre je je n'ai accès qu'à une observation, typiquement dans ton environnement t'auras l'état en entier tout l'état de ton système. Par contre l'État de ton agent. C'est pas nécessairement l'État en entier, c'est juste mes muscles sont dans telle position, je suis à telle position dans l'espace, je regarde à tel endroit et j'ai telle idée en tête. Quoi, tu vois ? C'est ça la différence. Oui, du coup on est sur une observation totale de l'environnement et non partiel. Ça dépend. Là ici, qu'est ce qu'on a comme information ? On a juste s donc s il peut contenir plein de choses et ça dépend en fait de l'environnement qu'on va coder typiquement dans le dans le labyrinthe c'est quoi le c'est quoi l'État ? Tu t'étais pas ? Non enfin qu'il y avait dans s c'était quoi concrètement ? Dans le labyrinthe, c'était le l'État d'avocat se trouve ouais et c'était quoi l'État ? Notre position ? Ouais, donc c'était juste 2 cordes. Donc là mon état c'était juste 2 chiffres, là mon état actuel ce sera un peu plus que 2 chiffres parce que il y a beaucoup de choses à modéliser. Mais un une voiture par exemple, son État, ça va être. La, le le, le comment dire, la direction de ces roues, sa vitesse actuelle, la position sur la route, sur quelle voie elle va se situer, et cetera. Et en fait, ça dépend de chaque système. Et c'est ça, pour le coup, un des premiers réflexes où vous devez avoir quand vous tombez sur une méthode de renforcement permis, c'est c'est quoi mon État ? C'est comme les actions et en fait, comment est-ce que je fais pour apprendre vraiment des choses ? Parce que ça, on va pas se comporter pas. Euh donc on a peu divergé on était sur le Monte Carlo avec cette idée que Ben si je peux me permettre de faire plein plein de d'épisodes, bah à la fin j'aurais une bonne estimation de ma valise fashion. Donc. Le le, le premier parodique classique, c'est exactement ce que je viens de vous décrire. Donc on fait. Pour estimer donc là on a presque un algorithme ici on quand on arrive sur un État s. On compte le nombre de fois qu'on a visité cet État là, OK ? Et on une fois qu'on a terminé l'épisode ? On met à, on met notre gain à l'intérieur. Parce qu'on connaît le gain à la fin. On met notre gain et on on sait qu'on a parcouru n fois cet État là donc on va faire la moyenne de toutes les fois que je suis passé par cette case. Et et du coup j'obtiens une estimation de mon m'a value fonction ce qui veut dire en effet que là dans mon cas où je fonce vers la porte je vais avoir une estimation sur toute la sur toute ma trajectoire mais je vais jamais avoir une estimation sur les endroits que j'ai pas visité ce qui est normal. Ok, et donc la loi des grands nombres ne dit que pour un nombre suffisamment de, enfin pour un nombre suffisamment grand de d'essais dans mon environnement. Eh Ben je sais que cette valeur là V qui est simplement la moyenne de mes de mes résultats sur le nombre de fois que j'ai que j'ai. Sur le nombre de fois que que des visiter la case va converger vers la valeur réelle de vie, c'est clair pour tout le monde ça. C'est vraiment le truc le plus naïf que vous puissiez faire, mais ça marche. Maintenant. Euh. C'est vraiment exactement la même chose mais ça vous le faites pas online en. Ensuite ? En pratique, comment on fait ? Pour. Pour calculer une moyenne quand on quand on ajoute de manière incrémentale. C'est ce qu'on appelle la méthode de Link Emmental bin OK. Ce qui fait que. On. On va le programmer comme ça à la fin. Il faut juste l'idée c'est de comprendre pourquoi ce qu'on fait. Ok, la moyenne c'est ça. J'ai k, j'ai k sample. Et donc vous savez faire une moyenne, hein ? On dit, on fait la somme, on divise par le nombre de fois qu'on a. Le nombre de fois qu'on a calculé. Donc l'idée c'est de voir comment est-ce que quand j'ai k moins un sample et que j'ai un nouveau sample carrive comment je calcule ma nouvelle valeur ? Ok donc là simplement on sort de la somme. On sort de la somme et on calcule notre nouvelle valeur, comme ça à dire on prend notre valeur. Euh, on prend notre valeur est moins un. Et on ajoute un sur 4 pour la différence entre XK et la valeur précédente. Ok, ça c'est rendu juste des maths basiques, mais ça fait que. Là, vous avez une manière de programmer votre. De programmer votre moyenne parce que là dans le dans l'exemple d'avant. Dans l'exemple d'avant, il fallait que je retienne 3 choses, fallait que je retienne la somme et le nombre de fois par lesquelles j'étais passé. Du coup c'est pas hyper pratique, ça veut dire que dès que vous devez réévaluer votre fonction il faut refaire la division ? Vous avez 3 valeurs qui sont et on comprend bien qu'elles sont liées et que c'est un peu chiant de garder les 3 en mémoire donc là en pratique on va juste garder. Le nombre de fois que j'ai visité et. Et la vraie valeur ? Et avec avec ce ce fameux élémental Min là. Eh Ben. On voit qu'on peut. On voit qu'on peut la calculer tout en gardant en mémoire uniquement la la valeur précédente. Donc. Pour voilà pour monter le. Pour chaque État. C'est exactement, c'est exactement ce que je viens de dire. Je. C'est exactement le même algorithme que cette fois, on fait notre. On fait notre pardon, notre notre update avec avec notre ami. Hop. Et maintenant, nouvelles règles. Ici, j'ai un un sur n parce que j'ai. Ici là, j'ai gardé en mémoire le nombre de fois que j'ai évalué mon mon état. Et en pratique Insurance c'est juste un un nombre entre. Comment dire, c'est juste un un nombre entre 0 et un. Qui va dépendre du moment où je suis, où je me trouve dans l'épisode ? Et ça reste 15 pieds de retenir pour chaque État le nombre de fois qu'on l'a, qu'on l'a croisé. Et dans certains cas même, ça prend énormément de place, ça veut dire, vous devez retenir une valeur pour toutes les pour tous les États dans lesquels vous êtes passé. Donc en pratique. On va avoir une incrémentalement qui va être implémenté comme ça. Avec un Alpha qui va être fixé. Ok. Ça, ça veut dire quoi, ça veut dire que là ? Je vais vous le dessiner, ce sera plus simple. Là j'ai mon point sur n. Donc je vais vous dessiner la. La valeur du Alpha. Quand j'ai un sur Rennes du coup. Mon Alpha, il fait ça super bol. Hop là, et à chaque fois j'ai. Donc là je vais voir mon. On VT ? Là c'est mon nouveau simple, OK nouveau simple, c'est GT. On est d'accord ou pas ? Ouais, c'est clair. Pourquoi c'était le nouveau simple. On a monté Carlo donc on vient de finir l'épisode donc on connaît le le gain à la fin OK et donc ? En pratique ? Ça, c'était mon estimation précédente, ça, c'est mon gars réel. Donc probablement que la valeur est un peu entre les 2. Dans le cas. D'Instrumental Min. Bah si j'ai que 2 valeurs au début mon mon mon Alpha il vaut combien ? Disait que 2 valeurs mon Alpha il vaut. Un sur n ça fait 1/2 donc on fait la moyenne entre les 2 parce que j'ai que 200 pour moi. Donc ma nouvelle valeur, elle sera là. Ok. Et ensuite si si, ça fait 10 fois que je passe par cet endroit là et Ben en fait là. Je suis beaucoup plus confiant dans cette valeur là parce que j'ai déjà 900 places pour pour la Palestine. D'accord donc hop. Ça va être là, mais en pratique c'est ce que je fais exactement ici hein. Ça c'est juste une moyenne, une somme pondérée avec la somme de 111 moins Alpha et Alpha qui fait un donc juste. Je vais me retrouver quelque part entre les 2 et c'est la valeur de Alpha qui définit ce où est-ce quelque part ? Ouais. Ouais, ouais. Ok, et donc en pratique si vous faites une emmental Min avec une vraie valeur de moyenne. Votre Alpha, il va faire ça ? Moi ce que je vous dis, c'est que, en pratique. En pratique on s'en fout, il faut juste prendre Alpha comme ça. Ok, on le fixe et du coup on n'a pas à retenir le nombre de fois par lesquels on est passé par le. Par le par le point OK, et ça donc c'est moins précis en termes de Monte Carlo, donc on aura typiquement plus de preuves de convergence avec la loi des grands nombres, et cetera. Mais. En pratique, en même temps learning, il y a les conduire le votre amendement, il va changer. Et donc vous allez avoir envie d'oublier d'une certaine manière l'État d'avant ? Par les les États que vous avez vraiment y a très très longtemps, donc typiquement si votre Alpha c'est 1/2. Été et bah tout le temps. Vous avez tout le temps retomber sur la moyenne entre le nouveau gain et le précédent, ça veut dire que j'oublie carrément le le l'État qui vient d'arriver juste avant. OK si à l'inverse bon elle fa c'est 0 0. Enfin pardon c'est 99, je vais garder tout le temps quasiment. Ma valeur précédente ? Et donc je vais très peu changer ma ma vue, ma pardon, mon estimation de V. Ce qui fait que j'ai j'aurais une valeur fait beaucoup plus stable dans le temps, ça peut avoir des avantages aussi. Sachant que même si c'est 99 si justement j'ai 1000 samples. Bah 99 ça fait que je vais quand même oublier les les 100 premières à peu près. C'est clair qu'on est en train de faire là. Donc tout ça pour dire que en pratique incrémental, mais on va pas, on va pas trop retenir. On m'a dit de faire ce on va juste faire cette opération là on prend la valeur qu'on avait et on ajoute Alpha fois la différence avec ma mon nouvel estimateur et la valeur précédente. Ok. Regarder autre schéma en tête avec votre Alpha qui part entre 0 et un. Ça dépend pas du temps, mais ouais, ça veut dire que. Ça veut dire quoi ? Imagine dans ton environnement. T'as un truc qui bouge, genre un ennemi ou je sais pas, t'es sur frozen Lake là et t'as un trou qui commence à grandir parce que le la glace elle fond. Alors c'est un. Eh Ben. Au début ? Je vais avoir une valeur de la value function qui va être. Hop. Glace là. Tant trop que là au début. Et il va grandir, donc il va arriver là. Et donc toi t'as une trajectoire au début ou t'as fait genre ça ? Et donc là t'étais content, tu vois ? J'étais content. Mais ton trou, il a grandi, hop. Et donc là typiquement ici t'avais une bonne valeur de la value function. Parce que qui était passé par là il y a il y a 3, il y a il y a 2 mois, tu vois et et ça passe. Sauf que là en fait, tu refais cette trajectoire là parce que tu fais de l'exploitation et en fait là tu tombes dans le. Ah, t'es pas content, toi ? Donc tu fais pas update, t'as la solution ici là et tu vas dire Bah maintenant là j'ai un gain qui est. Qui est négatif ? Pas moins 10 peut-être dans l'eau donc je vais voir l update. Et peut-être qu'en fait, j'ai envie d'oublier le la première trajectoire que j'ai fait. Parce que le l'environnement, il a changé, il n'est pas stationnaire. OK. Et donc c'est ça, entre les non stationnaires, c'est que en gros. La distribution de travail fonction, c'est-à-dire l'ensemble des valeurs que peut prendre va changer au cours de ton épisode parce que t'as des facteurs extérieurs qui confèrent que c'est possible, avantageux de se trouver dans telle ou telle position. Ok. Et donc dans ce cas-là tu vas vouloir avoir un Alpha qui est un peu plus grand donc tu vas plus vite remplacer ta nouvelle valeur. Tu vas plus vite pencher vers vers ton disque. Entre 0 et un. Moteur de choisir un pas de fois plus petit c'est ça ? Parce que là tu vois si tu veux enfin 0 Bah tu changes pas. Ouais Alpha égal égal un tu changes. Alpha y a un t'oublies carrément l'état, l'État juste avant. Et tu tu concernant ta nouvelle ? Ce qui peut être pas débile dans certains cas, toujours le faire intelligemment ouais. Euh mais si on a un épisode on peut se dire bon on peut mettre 140040. 0 bah là tu vois si t'as t'as bien compris. Tu vois que si ton Alpha vaut un sur K c'est équivalent à oublier les après les 4 derniers épisodes. Parce que là on a vu que si Alpha c'était 1/2, tu prends la moyenne des 2, ce qui fait que t'oublies la valeur d'avant d'une certaine manière c'est pas c'est pas, c'est pas rigoureux comme truc mais. Alors que si tu prends Alpha 1/3. Ataille à l'entière. Du coup, tu vas être ici là. Tu vas ici donc c'est un peu comme si j'utilisais aussi la valeur V de St. OK. C'est c'est, c'est pas très précis mais. Petit plus, on va être conservateur et je pense qu'il faut revenir sur un dessus. Envoyer avec ça, ça marche, merci. Ok, bon bon point Clémentine. Et après ? On termine ça, on fait quoi, ça vous va ? Hop. Donc. Ce qui me rappelle comment est-ce qu'on écrit la action value function ? C'est comme. Comment, du coup, l'espérance ? 0 plus gamma somme de non enfin oui si tu veux mais plus simple je pense les gains. Ouais sachant. Ce sachant s de +1 non s juste s ça ça va pas le fonction et là c'est action value function. S et a une fonction qui se pas ne prend plus uniquement l'état que l'on fait, mais l'État et l'action. OK, ça permet de choisir une action préférentiellement pardon. Et et on a notre comment cette action value fonction V de SA non ? Ouais. Je vais laver la main, Ah, tu sais c'est quoi la lettre qu'on utilise pour noter la canalisation ? C'est bien ? Ok. Et donc là. Pareil, on peut faire une une estimation de Monte Carlo. De la fonction Q directement. On fait la moyenne de tous les de tous les gains qu'on observe. Et on se rappelle que pour toutes les fois où on État s'était s ou l'action mon action CA et Ben je calcule ma ma fameuse moyenne où je le fais avec l'incrémenter quand on va devoir OK ? Alors petit point, mathématiques, on va pas, on va. Je vous épargne la preuve, mais c'est un truc important à savoir. Qui est que ? Là on fait l'estimation de la de la value function V de l'action value function Q. Ils sont-ils sont indexés par la politique. OK donc ma ma value function elle dépend de PI. C'est à dire qu'en fait. Là on se retrouve un peu dans le dans le premier problème qu'on a vu il y a 2 3 semaines du coup. Ou. En fait, je j'estime la valeur d'une policier. Mais le but c'est de c'est d'améliorer la policie. Donc si je rechange la police, il faudrait que je régime la valeur de la politique. Du coup c'est pas très intéressant. Et donc donc à priori ça va coûter encore plus cher. Parce que si il faut que je fasse plein d'épisodes pour bien estimer ma fonction, que en plus après j'ai d'être ma belle ici et qu'il faut la réestimer enfin faut estimer la valeur avec cette policier, ça va prendre vraiment énormément de temps. Sauf que non, parce que. Il y a la notion d'importance sampling, qui est aussi à connaître, pas forcément à connaître par cœur, mais l'idée c'est que en fait, si vous avez suffisamment exploré votre environnement, donc typiquement avec une Random policy. Eh Ben vous allez avoir assez d'informations pour évaluer n'importe quelle autre police. OK. Donc ça s'écrit comme ça mathématiquement, mais en gros l'idée c'est que si vous rechangez en gros. Grâce à votre nouvelle policier par rapport à votre baseline, donc là typiquement le B, ça va être. Votre homme, policier. Eh Ben vous allez pouvoir recalculer un estimateur pas trop dégueu. De votre alimentation pour votre nouvelle fonction ? Ok, intuitivement je dirais que c'est un peu comme si vous avez exploré tout le labyrinthe. Et vous savez globalement la distance, la distance au but. Et donc là, vous vous sentez un peu intuitivement, vous avez suffisamment d'informations pour évaluer une politique qui serait optimale. Ok, un peu comme ce qu'on avait vu dans le dans le Daming programming là ou au bout d'une itération en fait j'avais pas les valeurs exactes mais j'avais suffisamment d'informations pour trouver la police optimale. Ok, donc ça ça nous rassure parce qu'on se dit Bon c'est pas grave, je peux faire du Monte Carlo Sampling sur du Random avec une police Random. Et avec ça j'aurais quand même suffisamment d'informations pour pour évaluer alors bon estimateur de la police optimale et donc trouver la policier optimal. Ok, Ouais, on prend pour ça la politique qu'on a actuellement, enfin celle qu'on a trouvée par exemple dans dans le cadre de Monte-Carlo, on prend la politique, on a actuellement et pour évaluer la nouvelle policier, on a, on utilise cette année, c'est ça ? Donc typiquement là VOTRE B donc c'est votre hiver policier, ça va être un truc en homme si vous voulez tout explorer. Attention c'est ça, baissez votre exploration et maintenant vous allez vouloir maximiser ça en fonction de PI. Donc typiquement. Une fois que t'as t'as ton estimateur là parce que du coup ça veut dire que ça c'est n'importe quelle physique quoi. Tu vois, t'as t'as fait ton sampling avec B et ensuite tu peux évaluer n'importe quel policier. Ça veut dire que sans, sans avoir à faire aucun épisode, évaluer une polycopie. Donc du coup tu peux en tester plein parce que t'as pas à refaire des épisodes ? Et tu peux trouver celle qui va maximiser. Des enfin, maximiser pour tout, pour tout. Non, B C'est. La police avec laquelle tu fais monter carlo ? Ah, c'est la policier trouvé grâce à mon non. C'est la politique que t'as utilisé pour faire monter Carlo. Parce que monter Carlo, tu as besoin d'épisodes complets donc t'as besoin d'un policier pour faire monter Carlo. Policies, enfin, c'est tu Monte Carlo, tu fais un épisode entier. Pour faire un épisode en entier, il faut une policier. Et donc là ce que je te dis c'est que si tu fais une politique qui est suffisamment exploratoire, donc brandom typiquement donc là B C'est Random. Euh alors tu auras une estimation suffisamment bonne pour pouvoir évaluer n'importe quelle autre policier. Typiquement si tu te mets tu dis Ah bah tiens j'évalue la la policier optimale. Pourquoi tu vas te rendre compte que la vie du function là elle est très haute ? Tu veux dire Ah bah c'est mieux Random ça OKYA une question de l'air c'est bon ? Donc voilà l'idée que faire d'un work Monte-Carlo quand vous faites votre évaluation, on évalue une politique particulière. Et ça a priori c'est très limitant parce que la police on l'a connaît pas. Enfin la police optimale on la connaît pas à priori. Par contre les partisans shopping ça nous permet de justement de trouver cette police là à partir d'une évaluation non biaisée de la polyandre. OK. Donc. Pour ça, vous retenez l'idée d'importer du shopping. Là, vous pouvez retrouver les. Vous pouvez retrouver et donc là ce que je vous dis c'est, on prend aussi évaluation, on prend. On prend une biver de Random policy. Et en utilisant le l'importance sampling, on peut évaluer une n'importe quelle police TARGET, puis. Et à fortiori, on peut-on peut imaginer trouver la policier optimale, ça San, l'estimation qu'on a de. De la value function avec mon écran. Ok, pause, merci. On se dit, je préfère faire quoi ? On préfère 3 pauses de 10 Min ou. De 15 ? Ouais. De 15. Vers 10h45, Ouais, très bien. Donc je vais reprendre ici, ouais, ouais. Hop et je la reprends page à Ester mais je je le fais pas sûr toutes les cases en même temps je fais juste sur une trajectoire. Ok donc on va vite, on peut faire plein de bébés mais potentiellement si j'explore pas assez, Ben en fait je pourrais jamais rétro propager ce que je veux sur sur l'ensemble de l'environnement. Hop. TIC TAC, TIC et. Donc qu'est-ce que Ouais ça, ici on avait le Van, d'accord ? On reconnaît R +1 plus gamma V 2T+1 ça, c'est l'estimation. De V, 2T, on est d'accord. Ouais. Moins l'estimation à l'instant d'avant. Et ensuite on fait pareil, on fait un écran mental View ici on met notre Alpha avec notre et on ajoute ça à la valeur précédente de St. Ok. Donc ça, typiquement, vous pouvez déjà le programmer. Et ça vous permet de rétro propager le. Votre valeur à chaque fois ? Je. Qu'on a vu précédemment exactement c'est exactement la même chose. Simplement j'ai remplacé GT par RT +1 plus gamma VT +1. On est d'accord et du coup in fine j'estime la même la même chose. Simplement là je peux le faire pour toutes les. Alors que avant je devais attendre la fin de l'épisode pour le faire. Donc là je peux le faire petit à petit quoi. Donc même si j'ai un épisode qui est très long, Ben en fait si mon agent il explore suffisamment bien tout l'environnement, je vais pouvoir grapiller suffisamment d'informations partout. Tac TAC. Ouais, est-ce que c'est clair pour tout le monde ce qu'on est en train de faire ? Ok. Gardez bien à l'esprit que c'est vraiment la même chose sur ce qu'on a fait pour le dernier métro Programming, sauf que ici, on le fait sur une seule trajectoire. Et donc c'est c'est pas si évident que ça dans dans le cadre de l'approbation des autres, on était certain qu'on avait, on allait tomber sur le résultat parce que en fait la les l'équation 2 petits réalités de Bellman disait que. Si je connais pour pour tous les États, en fait, je vais forcément arriver. À l'optimum là en fait. Bah on dépend forcément de notre politique. Si on a une politique qui explore rien du tout, on aura une estimation sur la trajectoire qu'on a. Mais si jamais on n'arrive pas à la récompense finale, Ben on aura pas l'estimation qu'on veut, OK ? Mais en pratique, on on va bouger les curseurs d'exploration et cetera, pour faire en sorte que notre algorithme là converge bien. Donc exemple sur le cas d'un marche aléatoire, pour vous aider on part de on part d'une de ces cases là. Ici c'est une caste terminale mais dans laquelle on meurt. Enfin on a, on n'a aucune récompense et nous notre but c'est d'obtenir une récompense de un en allant tout à droite, OK. Et on va voir. Voilà, du coup on a une politique robe d'homme, c'est-à-dire qu'on va on va aller aléatoirement à droite ou à gauche. Ok. Donc là on va, on va comparer un peu comment vont se comporter des des algorithmes de type Monte-Carlo ou comme point. Donc ça c'est la la vraie valeur de la value function. Ok. Donc ça je peux l'obtenir de différentes manières donc en contre à peu près clair pour tout le monde. Quand je suis au milieu j'ai une chance de 0 5 de finir tout à droite ou de finir tout à gauche quand je me rapproche de l'objectif. Et Ben ma value function va augmenter. Et quand on se rapproche de de de la case de où on perd, bah forcément la fonction va diminuer. Donc ça, c'est ce qu'on cherche à obtenir à partir de. À partir de fin avec nos actes encore une fois, ce genre de cas-là vous le. Enfin vous le résolvez en 3 lignes avec de la programmation dynamique. Parce que y a très peu d'États. Mais là, l'idée c'est juste que c'est un exemple pour montrer comment se comporteraient les algos de de temple Defence ou de Monte-Carlo. Donc. Dans les 2 cas. Ou commence avec une estimation de 0 5 partout OK et on va faire des épisodes ? Toujours mettre à jour notre notre value un chat dans le cas du TD Learning pour chaque nouveau saut et pour chaque nouvelle récompense qu'on vous rend. On va faire notre update en pratique là on a une récompense de 0 à chaque fois donc il va pas se passer grand-chose tant qu'on arrive pas au bout. Ouais d'accord. Et là pareil pour pareil pour le Monte Carlo. Sauf que cette fois je fais mes updates uniquement quand j'arrive à la fin. Je termine mon épisode. Donc là, typiquement. Si j'ai un épisode qui termine ? Où j'arrive à enfin j'arrive au à la récompense. Pour t'aider learning bah je peut être uniquement l'état d'avant parce que je viens d'y passer OK ? Et pour monter Carlo Bah là j'ai update mes 3. En même temps, parce que j'ai attendu la fin de l'épisode. Mais. Comment dire ? Dans ce cas là ? Euh bah du coup l'exemple l'exemple un peu particulier parce que c'est on a un peu une une meilleure propagation dans le dans le cas de Monte-Carlo. Parce qu'en fait on fait tout d'un coup, mais faut bien imaginer que c'est aussi un peu spécifique au fait que on a une seule récompense à la fin. Donc dans ce cas-là en effet, c'est quand même plus intéressant de Monte Carlo parce que on va pouvoir avoir une vue plus d'ensemble là où le teasing c'est uniquement l'étape d'avant. Donc là c'est le genre de valeur que vous que vous Obtenez. Donc vous avez tout en haut la vraie valeur. Les on, on a détend avec Tom Ford, différents, on me détend avec Monte Carlo, on a des valeurs sont très proches. En pratique pour ce genre de de de pas de test, ça marche. À peu près pareil, comme on peut voir. Et. Ça vous donne un peu aussi un aperçu de. Concrètement, ça donne quoi ? Ici vous avez vos États, ici vous avez vos vos valeurs estimées. La, la ligne, la ligne droite, la la, les valeurs réelles. Et vous voyez à quel point les les valeurs, elles vont converger vers cette valeur finale. Ok, au fur et à mesure des épisodes que vous faites là, vous avez pour un épisode. Vous voyez pour tel poil de chance un épisode, il y a uniquement le dernier qui change et puis petit à petit, ça va venir se coller aux valeurs. Alors maintenant ? Ça ? Comme on a vu pour faire marcher ces algos là on a dû faire du Random. On avait une une chance sur 2 d'aller à droite ou à gauche, n'importe quelle que soit la méthode. Le le problème avec ça c'est que si vous avez un agent qui fait vraiment que du Random. Vous allez pas ? Vous allez pas pouvoir exploiter du tout ce que vous connaissez donc vous allez pas pouvoir aller très loin dans les épisodes. En pratique dans la plupart des des jeux ou des des environnements, si vous faites n'importe quoi vous allez rester juste à côté de du départ et vous allez jamais vraiment explorer votre environnement. Donc si vous voulez continuer d'apprendre, il faut un minimum d'exploitation. C'est à dire il faut être un minimum bon pour aller explorer un peu plus loin et récupérer les informations. Donc on va faire des choses un peu plus intelligentes, des un peu plus intelligentes que l'homme. Donc typiquement. On va là, je vous donne quelques ce qu'il faut noter. En gros, c'est le la dénomination. Là c'est épisode gladys. Je pense que la plupart d'entre vous devez connaître ce qui peut expliquer aux autres spécialistes, vas-y. Ouais. C'est un c'est un ajouter pour Ben savoir si on va exploiter donc si on va utiliser toutes les connaissances qu'on a acquises jusqu'à maintenant notre value. Bonjour ouais ce qu'on a jusqu'à maintenant et une aléatoirement faire. Bah justement une action au hasard pour bah avoir de nouvelles opportunités et potentiellement trouver mieux c'est ça ? Donc en pratique on met une une. On met une probabilité epsilon sur toutes les autres actions. Ça s'appelle excellent. Et sinon je choisis la l'action. Très mal donc. La la relate, oui. Où là c'est on va donner une probabilité proportionnelle à l'estimation de la value function qu'on a à ce moment-là. Ouais. La probabilité de faire l'action AI, c'est proportionnelle à. La l'action value fonction que je connais ne cet état là qu'on a. Que je normalise bien entendu par la somme des des actions. Ok, c'est assez légitime aussi. On pourrait la version glouton. On Pour rappel c'est juste on prend le on prend le Max. Donc là on choisirait choisirait un directement parce qu'on sait que c'est celle qui a la plus grande action ? Mg. Là on se dit, on veut quand même explorer et on va quand même explorer préférentiellement. Les. Les actions impossibles, typiquement dans le cas de crédit. Là, si jamais on a 2. 2 actions qui sont en fait potentiellement pas si mal. Bah là même la 2e elle se trouve aussi epsilon à côté de celle-là qui est à 5 qui est vraiment mauvaise. C'est donc c'est un peu bête, là on fait un truc un peu plus intelligent. Où on fait quelque chose qui est proportionnel à la valeur ? Qui a vraiment cette action ? Ok. Pas très compliqué, mais c'est bien de savoir que c'est possible de l'implémenter le jour où déjà, oui, pardon, l'invitation. Donc. 3e, 3e possibilité en utilisant softmax. Quelqu'un sait ce que c'est, un softmax ? C'est utilisé dans quoi ? Dans les Transformers, oui. Dans les Transformers, certes. Ouais. C'est ça, sauf que ça permet de transformer un ensemble de logit ça que moi qui en parle dans les Transformers. En probabilité. Quelle que soit la valeur de ces. De ces logs. Et là je l'exponentielle permet de vraiment mettre l'accent sur sur une certaine valeur là où ce qu'on a fait avec la roulette. Une qui marche aussi et ça donne trop d'importance aux résultats très faibles. L'exponentielle vraiment d'écraser les résultats des résultats qui sont même pas en booster des. Donc en pratique quoi, c'est c'est le plus utilisé ? Et en plus on va pouvoir paramétrer cette exponentielle là avec un bêta qui est la température. Je sais pas si ça vous parle de notion de température dans le sampling. Ouais bon je vais t'inquiète, il va tenter des doutes un peu exactement quand la quand la température tombe vers plus infinie tu vois tout tend vers 0 en fait ton PC va tendre vers vers la la loi uniforme. Ouais OK ça exponentielle de 0 donc un. Et là tu auras un sur la somme des des des actions possibles, donc un sur probabilité loi uniforme sur. Et contre la température. Et très très basse. Moi ça va booster encore plus mon exponentielle. Donc là vraiment ça va complètement écraser le les les peu de ces résultats pour avoir uniquement le progressif. Concrètement, ce qu'il faut retenir pour la température, c'est que quand t'en vers 0. Votre modèle va devenir déterministe. Décanter, dans vers plus infini, votre modèle va être complètement uniforme, ce qui est logique. Enfin ce qui est en lien avec la température, et ça devient très profond avec une notion d'entropie, et cetera. Mais c'est ça de rencontre Ivy. Mais atypiquement la température, c'est une notion que utilisée dans la génération de langage, vous allez pouvoir dire votre modèle. Ben tu génères de manière déterministe. Tu prends vraiment le mot qui est le plus probable avec ce qui a précédemment. Ou Ben là tu mets une température de 50 parce que je veux que tu puisses quand même un peu créer de nouvelles choses si jamais que tout. Et quand je te pose une question, tu me poses pas forcément les mêmes. Enfin tu me donnes pas forcément les mêmes options. Ok. Donc ça, c'est pour. La notion de. Pardon, les des outils d'exploration que vous pouvez utiliser. Et là je vous donne un petit un petit visuel de c'est quoi la différence entre ce qu'on a vu pour moi pour le moment ? Monte Carlo, on prend l'épisode en entier et pour chacun des détails de notre épisode on va faire notre update OK donc là on a une estimation qui est non biaisée parce que j'attends la fin de l'épisode pour terminer. Ensuite ? Là on attend pas la fin de l'épisode et on rétro propage de c'est +1 jusqu'à rester et ensuite pareil c'était plutôt sur rester +1 et cetera et là potentiellement on est biaisé vu qu'on a pas accès à toute la. Et enfin, ce qu'on appelle précédemment, c'était ça. Le dernier Mec programming où là on regarde tout en même temps. C'est le mieux, c'est ce qui marche le mieux qui va le plus vite. Mais en pratique, c'est souvent impossible dans des cas pratiques, hormones, cas dans des cas réels. Ok, maintenant qu'on a on a bien compris les notions de temple défense et de monter Carlo, on va pouvoir voir concrètement les les les algorithmes qui se sont utilisés à partir de ces notions là. Et donc. Le premier. Donc ça c'est un truc vraiment à connaître par ça. OK, il y en a qui connaissent déjà ça, ça leur dit quelque chose ou pas du tout. Ok donc là c'est vraiment votre premier algo de RL qu'il faut qu'il est encore utilisé un peu dans la vraie vie. On va voir et c'est très simple mais. Mais ça s'appuie sur tout ce qu'on a vu d'avant, OK ? Donc, dans samsa. On est. On est sur une approximation de ma de mon action value function OK et on utilise une politique Epson grise. Ok. Et on va voir que on va. Grâce à notre policier fun jeudi, on va pouvoir explorer un petit peu, récupérer de l'information et itérer comme ça sur notre fonction Q jusqu'à arriver à notre coq optimale. Donc. Pour chaque nouvel État. On fait du temple defence. OK sur notre fonction. Donc ça c'est la fonction D Update pour ça. Ce que vous pouvez voir, c'est. C'est pour ça que s'appelle Salsa, vous avez SAFSA ça ça ? OK. Donc ça c'est le petit le petit moyen mémo technique donc ça ça vous avez besoin de de LC à CRT +1ST+1AT+1. Donc ça fait ça et c'est pour ça que vous savez que. On modélise bien la fonction action value et pas uniquement la value fonction. Parce que vous avez besoin de l'action AT +1 aussi ? Et donc là vous avez juste à écrire belma. Normalement ça vous pourriez l'écrire par cœur. On fait une estimation de de Q de STAT avec l'information qu'on a de Q de St +1+1. Donc on est \*\*\* comme RT +1 plus gamma, plus de ça d'accord là-dessus ça c'est Batman. Et on fait une une Quentin sur notre notre valeur. Donc on prend la valeur initiale et on fait plus Alpha fois la différence on a trouvé OK donc là on est exactement dans ce cas de figure là. Sauf que on a ici QFAT et ici la nouvelle valeur. Qui est. R +1 plus gamma qd c'est +1. Vous félicitations, vous avez. Vous avez appris votre premier algorithme de l'information clary. Donc là c'est euh. Là c'est la version pseudo-code quoi avec le de qui est ici, mais ici vous pouvez, c'est juste vous instanciez votre modèle et cetera. Et donc ce qui a à retenir pour ça, oui. On l'a, on l'a dit tout à l'heure. Ouais. Et qu'est-ce que c'est que la définition de l'étude ? Comment on est dans un bar ? Parce que là c'est donc c'est l'attribution enfin de d'un État et d'une action à une récompense. Un petit peu. Ouais, mais il faut une définition mathématique pour mieux. Sachant S et a exactement. Pour rappel, ça c'est le truc que vous connaissez pas comme. Que le. L'espérance. Bien. Sachant. Ça doit être clair. Donc, dans le cas de Samsa, on estime cette fameuse action value fonction. En sachant très bien que à la fin, j'obtiens un costard là. Et bah Custard, on peut directement en déduire pstar parce que pour chaque État, on choisit l'action qui maximise ma mon action. Max OK. Et donc ce qu'il faut retenir de sarenza, c'est que. On est on policy. Et. On fait du temple Defence Control. Ok. Parce que on utilise la policy en epsilon ingrédient. Simplement mon opinion, il va descendre petit à petit. On va faire de de moins en moins d'exploration, de plus en plus d'exportations. Et donc on est on pousse et on fait du temple, c'est-à-dire fait pas du Monte Carlo, on attend pas la fin de l'épisode pour update to nos États. On le fait à chaque étape, petit à petit. Tac, TAC TAC. Ouais. Souvent ce qu'on va faire dans ce genre d'algorithme là ce qu'on va faire baisser, puis c'est l'un de ces petits. Non, là on l'a pas marqué, mais on. Du coup ça permet d'avoir un truc qui exporte beaucoup au début et petit à petit il va faire de l'exploitation et donc il va aller plus loin. Donc on va avoir plus d'informations sur le sur la voie finale OK ? Oui, une mise à jour au niveau de l'itération suivante. Ouais ça ça marche que dans un cas discret hein. Si on est dans un jeux continue, on aura plus utiliser mon c'est une très bonne, c'est une très bonne question et ce sera pour les courses suivantes, donc la remarque très judicieuse si l'on déjà excuse-moi. Remarquez juste de d'arthur, c'est que pour ici, on est sur des États et des actions discrètes. Or dans la vraie vie, tout est continu, donc on a, on va avoir un problème quand on va vouloir, quand on va vouloir modéliser des des environnements qui sont continus. Et donc là, aujourd'hui, on est toujours sur du discret. Et. Et on va dans un de vos TP que je vais vous donner juste après, là on est sur un environnement qui continue. Et vous allez voir qu'on a été obligé de discrétiser l'environnement. Pour pouvoir utiliser ce genre d'algorithme. Ok, il est très très bonne remarque, très déçu oui, le R proposant de quoi en fait le R le grand R ici. C'est ce qu'on observe. L'observation. Ce qu'on observe c'est et s prime, donc ça veut dire quoi ? Ce que nous donne l'environnement contre une observation, une réponse, ouais. Ça c'est très clair, hein. Toutes les lettres et toutes les lettres sont au tableau. Ça doit être clair pour tout le monde, hein ? Ouais. C'est pour ça que j'ai beaucoup, j'ai beaucoup à dire, insisté là-dessus dans les premiers cours. Là on se retrouve dans des dans des cas où on écrit des formules. Si vous savez pas ce que c'est les lettres que je note à chaque fois, il faut se poser des questions. En fait si vous allez juste pas comprendre que en fait là QSAA prime et cetera, normalement tout est clair. Ouais. Vous avez le droit de poser des questions, c'est même c'est pas clair et il faut les poser très sinon vous allez vous allez, vous allez partir. Ok. Maintenant 2e algorithme que learning OK cette fois. Vous voyez, ça ressemble beaucoup, hein ? On a on observe SARS où on aurait pu l'appeler Sarse. Mais on observe pas assez +1 enfin on observe pas, on le prend pas en compte OK ? Pourquoi ? Parce que ici là on a plus acheter +1 on a un axe. Ce que tu peux m'expliquer que sont en train de faire là ? Ouais justement alors on on a l'action AT mais on on l'utilise pas pour faire notre update. Tu vois avant c'était ça SARSA Sara ce qu'il vient de faire et là on utilise pas a de a de t +1 on utilise à la place on met un maxi. Du coup, genre fondamentalement en train de faire quoi là ? Fonction, joueur d'action un peu élevé pour déterminer l'action qu'on prendra après. Ouais. D'une certaine manière, oui, Regardez. Plus grande valeur de plus pour l'état suivant. Ouais exactement. Ouais ouais, on ce qu'on est en train de faire, on regarde le maximum des valeurs de fin, de la meilleure des actions possibles. Ouais c'est ça. Et donc fondamentalement ça veut dire que dans le dans la situation précédente. Notre policier c'est un ingrédient. Donc si jamais par erreur. Enfin parce que c'est du Random, on a fait une action de \*\*\*\*\*. Ben on va update notre notre value function non par notre action value function. Pardon on va on va faire une update avec cette valeur pourrie qu'on a choisie. Et ça, c'est pas ce qu'on veut. Nous ce qu'on voudrait c'est que directement on apprenne la la la fonction l'action value function. Qu'il soit optimal. Bon d'accord, nous ça nous intéresse c'est de trouver le le pistard à la fin. C'est pas on s'en fiche d'avoir un. Du coup en fait ce que je dis c'est que en fait j'ai déjà l'estimation du Q de S pour toutes les actions possibles puisque ça fait je disons 100 épisodes que je suis en train de faire mon de faire mon pull en hein donc en fait au lieu de faire mon update avec l'action que j'ai vraiment choisie qui est potentiellement mauvaise parce que je suis encore en train de faire exploitation de l'exploration, pardon ? Mais en fait, je prends directement la meilleure valeur possible. Donc ça, ça veut dire que là je suis directement en train de calculer l'action value, fonction associée à la politique optimale. Ok, du coup est ce que ça serait pas ? Ça nous empêcherait pas de trouver des potentiels meilleurs chemins parce qu'on prend le meilleur chemin qu'on a vu jusqu'à maintenant. Donc c'est la subtilité. C'est que c'est pas parce que tu suis une politique, une crédit. Que tu peux pas calculer ça. Parce que en réalité là ça c'est valable quel que soit la plus propre et c'est pas parce que je fais mon B comme ça que je suis obligé de dire que à tes +1 ce sera le bac. Un oui, oui, je veux dire. C'est à dire en gros là dans mon dans mon graphe, là si je remonte. Là je suis là-dedans. Disons que en fait, ça, c'était l'action optimale à faire. Pardon ? Imaginons que donc les les actions c'est des points là ? Imaginons que ça, c'est ma valeur optimale, donc. Comme comme je veux avoir une estimation de mon action, enfin de ma fonction Q. Je vais aller propager cette valeur là sur mon état précédent toc pour aller jusqu'ici. Par contre ça veut pas dire que je vais pas essayer d'explorer cette branche après, donc j'utilise l'information que j'ai au maximum même si je choisis d'explorer une branche différente de l'arbre. Ok. Donc ça s'appelle le cul learning, parce que dans le cul learning vous essayez directement d'apprendre plus tard. Indépendamment de la politique que vous suivez. OK. Et donc c'est pour ça que s'appelle of policy. Commencer à comprendre pourquoi est ce que on pose ces mots à chaque fois sur. Sur les algues qu'on fait dans ça, ça on est on policy parce que. On dépend de la politique, on est en train de suivre, donc fondamentalement on dépend de AT +1. Dans le cul learning. On peut directement le Dead, c'est optimal en supposant qu'on fait la policier optimale, même si on est en train d'explorer qu'une politique est différente. Donc on est off progressive, OK ? On n'a pas besoin de connaître un peu plus. PAF, PAF, PAF donc la politique. En pratique, on va toujours faire des trucs du style Épicène gradi ou soft max ou j'en sais rien. Mais. Mais on converge quand même plus vite vers la fonction Q star. De laquelle on va pouvoir extraire nos policiers, peux OK ? Et donc il y a pour les matheux. Vous pouvez regarder, y a des preuves de convergence si on montre que notre policier explore suffisamment. Voilà tout, c'est un peu bas, c'est un peu le même preuve que ce que vous avez sur bellman pour le le cas dynamique. Sauf que cette fois c'est beaucoup plus fort parce que on connaît pas. Donc c'est c'est des des marches aléatoires dans dans l'espace de de possibilités. Vous montrez que si votre agent il explore suffisamment, et Ben vous avez quand même convergé vers la fonction custom. Donc c'est assez formel oui. On peut le le modèle dans le sens où par exemple il est dit, il va déterminer sa police en fonction de bah, de de pusta que c'est. Les meilleurs, la meilleure fonction actions, récompenses qui va avoir ? Ouais et étant donné qu'on a mise à jour tout le temps avec la meilleure récompense potentielle possible sans pour autant faire le CE choix là. Parce que ça va utiliser le modèle. Bah t'es toujours sur le tempo la défense parce que tu utilises d'accord hein, faut pas on fait pas de de Monte Carlo aussi donc t'auras toujours un biais parce que t'as toujours la notion de t'attends pas à la fin de l'épisode pour avoir la vraie estimation. T'as une estimation qui est que toi t'as mis au début donc en fonction de enfin tu as tout à l'heure a pris 0 5 partout. On va prendre 0 un, j'en sais rien donc en fonction en fait tu vas être biaisé dans un sens ou dans l'autre. Donc c'est pour ça qu'on a vu tout à l'heure là. On a vu c'était quoi, c'était à ce moment-là là, on partait de 0 5. Et donc on voit que on là ici, on va surestimer la la valeur ici et on va sous-estimer la valeur la. Et donc ça, ça et ça dépend beaucoup de l'état de l'Initial dans lequel tu trouves, mais c'est pas. Plus biaisé que ça, au contraire, c'est même, je dirais même, c'est un peu moins biaisé parce que dans sarsat à l'idée que si jamais par malchance t'as pris la mauvaise action. Bah tu vas rétro propager cette mauvaise action sur ton état précédent donc tu vas la \*\*\*\*\*\* vers le bas alors que dans le cul learning tu peux faire une action qui est qui est bonne. Pardon qui est mauvaise mais ne pas rétro propager cette mauvaise valeur dans ton graphe parce que. C'est que mon exactement. Et parce que nous à la fin, ce qui nous intéresse c'est d'avoir plus tard, tu vois ce que si, comment dire ? Si ton but c'est d'estimer le la, la fonction Q pour enfin même la la table des Q en l'occurrence pour ton épilogue grandit, il faut que tu le fasses. Mais nous comme ça vous intéresse, c'est de trouver l'optimum. Bah on s'en fiche parce qu'on en pratique ta police optimale tu prendras pas cette action là parce que tu sais que la valeur est mauvaise. Et donc là vous avez maintenant votre votre 2e algorithme que learning donc qui marche en pratique mieux que ça. Et ça encore une fois, ça dépend peut-être que vous aurez envie de pour ça. C'est du un policier. Donc si vous avez une politique spécifique. Vous allez avoir envie de, de, de de l'estimer, donc vous avez besoin de faire du Hand Policy, donc vous allez pas faire du cul, donc vous allez faire du sport. Donc là pareil, vous avez de quoi l'implémenter, et en pratique, là on aura 2TP et on aura un vous implémentez ça pour ça, et l'autre pour l'ordi. Sur 2 cas différents. Donc vous avez question là-dessus, est ce que l'algorithme est clair pour tout le monde ? D'où ça vient ? Pourquoi c'est un point defence ? Pourquoi c'est on policy ? Pourquoi c'est of policy ? Bah super je vais très doué si tout le monde comprend. Hop donc. C'est rapport à Belman. Dans le cas de Salsa, utilise l'équation de Bellman. Tel quel, où on utilise directement la la valeur avec l'action qui suit. Et dans le dans le calcul learning, on utilise la les cochons de bellman optimales. C'est-à-dire bellman écrite en sachant que la, la politique qu'on évalue, c'est la politique optimale, et donc on peut changer l'espérance du du cul là directement en Max les culs, pour toutes les actions possibles. Et bien rapide. Hop. Voilà, petit exemple. Ouais donc pour le cul learning ici donc là c'est c'est l'État où on a une petite voiture qui doit arriver jusqu'à la Coupe. On retient pour tous les États possibles. On va regarder toutes les toutes les actions possibles, donc là c'est dans quel cas se constitue. Et. Toujours cette idée de. Même si je choisis d'aller là. Et ensuite je choisis d'aller là-bas. Je change pas ma valeur qui sera ici parce que je sais que ici, si je vais à droite, je je, j'arrive, j'arrive. C'est ça ? Ouais, même si je même si je sélectionne l'action d'aller à droite. J'obtenais, une reward 0. Je peux faire mon update ? Mon nouvel État, c'est celui d'à côté. Je peux calculer mon. Je regarde dans tous mes États possibles. Ah je vois que là j'ai un un parce que je je je peux terminer l'épisode donc même si je choisis, je ne choisis pas cette action là. Eh bien. De qu'on peut rétro propager la bonne valeur et donc avoir une meilleure estimation que si, si on faisait du samsa, on aurait pris la la valeur de la de l'action qu'on aurait. Et voilà, oui, définir du coup l'activité table, ça passe dans un premier temps par explorer justement. De l'ensemble de l'environnement. Ouais. Dans dans ce cas-là, le tutoiement est. Là t'as les tous les États possibles, là t'as toutes les actions possibles. Au début tu vas partir de peut être de 0 partout, il y avait des zéros ici et tu rétro pages comme ça. Là je viens de faire une action pour aller de là à là et je connaissais la la la la la valeur de Q pour étant la et faire ça parce que je sais j'ai terminé une fois l'épisode. Donc là, je choisis d'aller là. Et je rétro propage directement avant même d'avoir choisi de faire la de la meilleure action possible. Je peux quand même rétro propager ma valeur. Donc ça c'est une meilleure estimation. 0 95 depuis alors qu'avant j'avais quoi ? J'avais 0 4. Et là on est dans un environnement qui est quand même qui est quand même restreint. Elles ont quand même qui sont beaucoup plus grands. De problème d'avoir de justement de pouvoir 7 l'ensemble de la cuve table. Bah si complètement là on accorde. D'accord, à 12, on va plus. Non, ridicule de valeur par rapport à ce que tu pourrais avoir. Faire des milliards de milliards de possibilité, quoi ? Mais c'est pour ça que on va rapidement tomber sur les limites de ce genre de modèle où on modélise toutes les valeurs possibles. Donc là encore, encore une nouvelle, une nouvelle. Comment dire visualisation de pourquoi est-ce que on parle de off pour les sims ? Mais j'espère que c'est clair pour tout le monde. Euh, notre notre. Notre billet dehors, c'est c'est faire du Random. Mais ça ne m'empêche pas de. D'estimer la véritable la véritable Poly pistard, même si c'est pas celle qu'on suit alors. Encore un nouvel exemple ? Oui, un petit point rapide. Il y a une autre. Une autre version de sarka on fait du Expectorer Sara. Où on va, on va utiliser belman. Pour faire une estimation au lieu. Au lieu de de prendre directement la valeur de de qu'on avait là. Par exemple, celle qu'on avait là on prend directement R +1 plus gamak de St +1+1. Et dans l'expectative, Sara. On rajoute encore un petit peu en. En se rappelant des épisodes qu'on a fait précédemment. Donc là le logiciel vous avez dû, vous avez dû Tom traditions, donc c'est à dire que une fois qu'on a fait un épisode, on l'oubli. Et dans dans l'espace texa. On va pouvoir faire une meilleure estimation en faisant le la somme de ce qu'on a développé. Ça, c'est moi, c'est moi d'utiliser ça. C'est bien de c'est bien de de se rappeler du nom est trop content. Qu'est-ce qu'on raconte là-dessus ? Ok. En résumé ? Nous, ce qu'on vient de voir. On vient de voir différentes différentes choses. D'un côté on a du full backup ou ce qu'on ce qu'on a dit tout à l'heure, c'est plutôt du du Monte-Carlo. On fait l'épisode en entier et on regarde pour chaque État à chaque fois comment ça se passe. Ou du simple backup temple divisions où on va uniquement avoir point à prendre, donc ça prend moins de place en mémoire, mais c'est biaisé, et cetera. Donc là pour ceux qui aiment bien les graves vous pouvez voir un peu la différence entre farsa ou dépend de la police. Donc on a uniquement cette ligne là et que learning cette fois on a notre action, notre nouvel État et là on va considérer tous les toutes les actions possibles derrière. Donc là on devient of policy. Euh. T'en fais pas et. Vous voyez bien qu'on est différent de ce qu'on a fait dans le cours précédent dans le Dynamic programming d'épée, c'est pour programming. Ou. À chaque fois, on considère le graphe en entier. Pas dans le graphe, il est petit mais c'est la, c'est la même commentaire. Si votre votre environnement est trop gros, vous pouvez pas en fait tout parcours, mais là on fait un full backup. On connaît tous les États, toutes les actions et toutes les toutes les valeurs qu'on. Et voilà, du coup, ce petit tableau, ça vous permet de. Faire un peu un parallèle entre les les algorithmes qu'on a vu dans le cas du dernier microgramme et les algorithmes qu'on vient de voir aujourd'hui. Où ? Au début, on avait l'itératif pour les évaluation où on avait une policy. On calculait la la, la value function. Ensuite on trouve une nouvelle policy. Que là on peut rapprocher du TD Largu. Où on a. On a un État. On fait une action, on fait un nouvel État et on trouve une une nouvelle policière à partir de ça. Donc on prend justement la différence entre ce que j'ai, ce que j'ai reçu ici, ce que j'ai réussi. Ensuite, on a le top politisation où je partais de ma première action. J'observe mon résultat et j'observe. L'action qui va suivre après ? Alors que là je j'observe uniquement l'État, on est du value base. Et là on observe aussi l'action, on passe de là, on passe de la value function, là on prend plus d'informations, on prend la on prend l'action value function. Et donc là on est sur du samsara. Parce que ça ressort. On considère notre État, notre action, PAF, on a l'étape, on a l'état d'après et on considère aussi l'action qu'on va faire après pour faire la la rétrogradation. Donc on peut le comparer à notre que policier estimation ou cette fois au lieu de d'itérer sur la la value collection. Pardon, les l'avait ici, mais on itère directement sur la police. Et enfin du coup ? Quand on utilise l'équation de bellman avec l'optimalité, donc, c'est le fameux moment où remplace. La valeur actuelle par le fameux Max Là. Ici, on est sur du curnier. Et là, on est sur. C'est ce qu'on avait fait avec l'algorithme de evalutation. Donc cette fois pareil, on prend pas uniquement. L'action qu'on va faire sachant la policier ? On prend toutes les actions possibles et on prend la meilleure. Donc on suppose d'une certaine manière la police. Ok donc là ça vous sert un peu à vous rappeler tous les animaux qu'on vient de voir. C'est quoi les liens entre chacun ? Ce qu'on considère à chaque fois ce que je considère la value function l'actu action action value function. Est-ce que est-ce que je considère ce que je suis of policy ou on policy du coup ? Donc voilà, c'est ça. Oui je sais pas qu'on pourrait quand vous avez expliqué le cas de Monte-Carlo dans cette configuration. Parce que là je vois la colonne dans l'Amérique. Pourquoi Bing et temporel du France ? Sauf que pour la colonne Monte-Carlo, je sais pas ce que vous avez dit. C'est peut-être un peu maladroit ce que j'ai dit, mais ce que ce que j'ai dit c'est que dans le cadre de mon tchao. Faire un épisode en entier donc c'est un peu plus proche de ce qu'on a fait dans le dernier programming où on considérait un peu tous les épisodes possibles en entier, mais en même temps quoi ? Parce que là, dans ces dans l'intelligence. On n'est que sur tu vois y a 2 ou 3 cellules dans ton graphe juste et sûr j'ai mon état j'ai mon État +1 j'ai des actions comment je fais en connaissant, en ayant un peu de connaissance sur mes actions, et cetera pour être propagé à raison après là où pour le bananier programming on le fait mais pour tous les États possibles en même temps. Ce qui se rapproche un peu plus de Monte Carlo ou on le fait pas sur tout le graphe mais en fait sur toute la trajectoire en entier. Je sais pas si c'est très très droit de ma part de faire cette de faire le. Hop. Toc, toc, ça pareil, c'est un résumé de toutes les. De toutes les fonctions de update qu'on a vu. Ok pareil sur le micro gaming. Comment est-ce qu'on a fait là ? C'était juste bellmann tout bête. Ensuite, on a rajouté bellman, mais avec. Avec l'action menée fonction et enfin bellman avec l'action value function, sachant qu'on est sur une police optimale. Ça, c'est pour l'approbation dynamique et là c'est l'équivalent en temporal défense où on n'a pas. Tous les États possibles mais on fait, on fait notre update petit à petit. Ok. On va bientôt passer le TP mais avant ça du coup, est-ce que vous êtes bien au clair sur ? Vraiment, ce qui est important sur ce qu'on a fait aujourd'hui, c'est les fameux algorithmes ici là. Ce que vous êtes bien au clair sur comment elle fonctionne, pourquoi est-ce qu'on en est arrivé là ? Euh, les notions qu'on manipule en dessous, et cetera. Ouais bah très bien hop. TC. Est-ce que vous voulez faire la pause maintenant ou est-ce que on fait une pause au milieu du TD ? En gros là j'ai 2 TP à vous faire faire. Et donc. Soit on fait une pause au milieu des 2TP, mais comme il y en a qui vont pas aller à la même vitesse, c'était plus pratique de faire de pause maintenant. Bah du coup, rendez-vous à à 40. Et pour ceux qui veulent commencer maintenant, je vous mets le lien. Je vous ai fait des cols. Du coup pour cette, c'était plus pratique pour la pour les installations, ce que tout le monde avait réussi à faire. Tourner les les notebook chez lui à la fois ou pas ? Ouais. Ok bon bah ça vous servira pour et là vous pouvez télécharger pour le parc local, mais au moins vous êtes sûr que vous avez pas de prince. Hop. Je vous ai mis les liens sur sur moodle, normalement on pouvait les voir. Merci Monsieur. Il s'est fait embarquer, lui.