Vrai que les paramètres c'est ça, ouais. Juste petit.de notation, d'accord. Donc. Pour paramétriser notre, notre politique, on a. On a beaucoup de manières différentes de le faire, d'accord, on. Là je vous ai juste rajouté les les contraintes qu'on a parce que on peut-on va approximer notre politique mais on peut pas, on peut pas l'écrire n'importe comment, d'accord notre politique. Donc là on a une politique qui est stochastique, OK donc ça c'est une probabilité. Donc ça ça veut dire que. Bah déjà toutes les. Toutes nos sorties sont positives, ça c'est logique. Et la somme de de la des probabilités de toutes les actions, ça doit faire un pour tous les États possibles. Ok, donc juste pour faire gaffe, on approxime, on approxime une distribution et donc il faut faire attention à. À ce que. Certains contrats. Ça, c'est un exemple. Donc la Poly softmax ? Si j'écris ma politique de cette manière-là OK bah là en fait on est sûr que elle va respecter le les contraintes qu'on vient de de soft mais ça vous parle soft est une fonction qui à partir de log enfin à partir de n'importe quel valeur transforme ça en l'autre réalité. Donc là ça veut dire que on se dit Bon bah en fait si je l'écris. Quelle manière-là j'ai même pas à me à me associer des contraintes en fait je sais qu'elles sont, elles sont respectées. OK donc là on apprend plus les probabilités directement, mais on va apprendre ce fameux h là. Et ce h qu'on appelle ça une préférence, je veux pas. Donc enfin ça action préférence OK donc on apprend une préférence ou des logs si vous voulez ça ça revient, ça revient à peu près au même. Et ensuite on va transformer ça en probabilité avec un soft donc une possibilité, on est obligé de faire comme ça. Bon c'est juste qu'on appelle ça la préférence, mais c'est juste que tu vois juste avant. Euh. On a on, on veut approximer la politique et on sait que la police c'est une une localité donc elle va respecter ces contraintes là. Donc si je l'écris comme ça. Donc là je l'écris de manière-là avec la manière assez Max. Donc là on va apprendre h plutôt que la pollution lancée, on donne à h le nom de préférence. Comme ça on sait que h peut prendre n'importe quelle valeur donc h chez une valeur dans R maintenant alors qu'avant on était en valeur dans 0 un. Et on sait qu'à partir de n'importe quelle valeur de h, bah on va pouvoir avoir une bonne probabilité. Donc là c'est c'est ce à quoi ça ressemble. Si on a des préférences comme ça avec un softmax vous voyez là on peut avoir des des des valeurs qui sont négatives avec un soft Mac. Bah on s'en sort comme ça. Donc là on prend quand même beaucoup plus en compte le fait que bah là cette valeur là elles sont à peu près équivalentes. Donc dans notre policier si je fais un softmax si on a à peu près la même préférence pour les 2. Alors on a quand même une probabilité non nulle de prendre la 2e action, OK ? Alors que par exemple, dans le cas des psychologues dédiés. Bah Epson Grey c'est quoi ? C'est je prends la la meilleure action avec avec probabilité un moins epsilon et toutes les autres actions avec avec le reste. Donc là ça veut dire que même si cette action là elle est quand même très proche en préférence de la première, on a quand même très peu de chances de l'apprendre. Donc c'est un peu moins intéressant, même si c'est quand même plus simple à à coder. Faire un sanglier plutôt qu'un. Donc voilà. En général, on va très souvent, on va très souvent faire quand on quand on fait le de, comment dire quand on évalue des policiers comme ça, on va très souvent le faire de cette manière-là parce que bah vous le savez, on vous avez dû le voir en masse, mais le fait de le faire avec des préférences, donc des valeurs R pour un réseau de neurones, c'est beaucoup plus simple à apprendre que directement à output. Quelque chose dans 0 un. Ok et donc ? Et le soft Mac ça ça passe très bien quand on à la vague propre donc donc c'est très bien OK ? Donc. On sait à peu près ce qu'on peut faire, OK ? Petit point rapide sur le fait, qu'est-ce que ça veut dire en fait de paramétrer notre policier, OK ? Euh. Globalement enfin, c'est les avantages par rapport à ce qu'on a fait jusqu'ici, c'est-à-dire plutôt apprendre une value function sur tous les États ou sur tous les États, les actions avec la Q function. Qu'est-ce que, qu'est-ce que ? Enfin, comment ça nous avantage de directement paramétrer la police ? Bah. Déjà on va voir, ça converge plutôt bien parce que, et je pense que l'intuition derrière, c'est que que l'intuition derrière, c'est que dans le cas de si vous voulez apprendre la value function, vous allez vouloir apprendre sur tous les États possibles alors que la police, si en fait elle elle va être capable. Enfin si vous apprenez uniquement une. La loi de de. De comment dire de comportement pour votre modèle ? Là, il va pouvoir tout seul ignorer les États qui l'intéressent pas. Et donc en fait c'est un peu comme si vous exploriez votre votre, votre espace un peu plus efficacement parce que je vais pas prendre sur les attaques sont pas intéressants. Là où la vallée Function Bah je leur ai demandé d'apprendre la value function aussi sur les États moins intéressants donc elle a un peu perdu de perdu de la comment dire de de l'efficacité là-dessus ? Donc ça marche, donc c'est un peu la même idée mais ça marche pas bien. Donc pour les problèmes de dimension et ça permet d'apprendre notamment des. Des des politiques qui sont stochastiques. Je les inconvénients. Pour la même raison, comme on explore directement, on optimise la policy. Bah c'est un peu plus facile de tomber dans une policy, donc on a moins d'exploration, donc c'est plus facile de tomber dans un dans un optimum local, là où justement quand on fait de la value function approximation on fait tout en même temps. Donc on est, on a, on est, on a. On a davantage de garanties que on va explorer un peu tout l'espace et qu'on va moins tomber dans un minimum local. Et. Et en gros, ça coûte cher. L'idée, l'idée c'est ça, c'est que. Comme on itère sur la politique, bah tant que le policier est pas bonne, on va pas pouvoir explorer suffisamment et cetera, donc ça coûte rien. Globalement cet inconvénient là c'est un peu un inconvénient du R quoi. Globalement l'URL, on sait que ça coûte cher. Euh OK. Petit exemple, pour vous donner rapidement plus des instructions sur ce qui va se passer après, on se place dans le cas d'un. D'un d'un collecteur de balles de tennis ? Ok, c'est un petit robot qui va attraper les balles de tennis sur un terrain de tennis. Mais il ne il n'a pas de capteur pour savoir où sont les balles. En gros ils se baladent et il a des récompenses quand il récupère une balle mais il les voit pas OK ? Donc, et on mesure sa performance comme étant. Le nombre de balles qu'il arrive à récupérer, c'est juste ici. Je pense que je vous perds pas, mais ce qui est intéressant c'est que dans ce cas-là. Le la récompense elle va juste dépendre de la trajectoire du robot qui est pas de sa enfin sa trajectoire en entier et et de la position des balles OK mais lui il a pas accès à cette position là devrait un peu indépendante. Donc nous ce qu'on va contrôler, ça va être la policier, OK, c'est à dire c'est quoi la trajectoire ? Préfère mon robot, qu'est-ce que je code à l'intérieur ? Il va faire quoi ? Est-ce que je fais un truc un peu aléatoire ? Tu fais du partout et ce que je fais une une trajectoire un peu optimisée où je sais que le modèle va passer partout avec des trous. Où vraiment vraiment passer partout. Mais du coup, une trajectoire plus longue qu'on va voir ? Peut-être. Il y a plus en général dans un terrain de tennis, il y a davantage de balles dans le fond du terrain, par exemple, peut-être avoir un tête à plus passer au fond du terrain, et cetera. Euh. Et donc ici on introduit la notion de police Search où on va bah faire la recherche de la meilleure politique possible. Ok, on. Ici enfin, c'est juste pour comment dire, pour vous montrer à quel point les les les, les 2, les 2, comment dire variables qui vont influencer notre résultat sont indépendantes. Je peux faire d'une part juste varier la position des balles sans faire varier la position, enfin la trajectoire de mon robot, ou alors faire varier la trajectoire sans faire bouger la position du bal. Donc ça va être 2 sources de 2 sources d'aléatoire, donc modèle. OK. Ok, donc comment est-ce qu'on fait pour pour rechercher la meilleure police ? A priori c'est dur, OK, donc trouver la meilleure politique. Déjà il faut qu'on se définisse un objectif. L'objectif ? On l'appelle J très bien. Et on le définit comme l'espérance du gain du Return. Donc là on est quand même. On est globalement tous d'accord, on veut optimiser le on veut optimiser le le gain total sur l'épisode OK et donc dans ce cadre-là on on l'appelle Utility fonction la fonction d'utilité. Ok. Et.de vue notation, donc là on commence un peu avec les maths, mais on J thêta, on peut l'écrire comme ça. Ok, donc donc je donc une espérance ? D'accord, l'espérance ça va être, ça veut dire qu'on va faire de moyenne sur tout, toutes les possibilités. Et cette moyenne là je peux la calculer sur enfin je peux diviser mon espace dans comme je veux. D'accord ? Et un truc qui va être intéressant aujourd'hui, ça va être de l'écrire comme ça, c'est-à-dire que je vais faire la somme sur toutes les trajectoires possibles. Ok donc parce que justement on est dans ce cadre-là en fait ce qui définit. Reward, c'est la position des balles et ensuite nous, ce qui va, ce qui va aussi définir ce qu'on va. Enfin les récompenses qu'on va obtenir, c'est la trajectoire qu'on a, d'où le fait d'écrire notre notre espérance comme la somme sur toutes les trajectoires possibles. OK, donc ça vous êtes d'accord que c'est juste ? L'espérance du gain et s'écrit bien comme ça, OK ? Qui notamment parce que il y a l'idée que en fait, pour avoir la moyenne de l'espérance totale de notre gain, bah il suffit de faire la moyenne sur toutes les trajectoires aussi, OK ? Donc donc avec la trajectoire possible qui est la la probabilité de la trajectoire, sachant qu'on utilise la polycopie en excès par thêta. Ok, donc on va vouloir optimiser ça, donc trouver le thêta star, donc le le thêta optimal qui va. Qui a maximiser notre espérance du retour ? Bien. C'est clair pour les nations. Ouais. Ok. Donc. L'idée ? L'idée globale de de ce qu'on va faire pendant ce cours là ? Vraiment le l'idée fondamentale, c'est ça. Sans faire aucun calcul, on peut imaginer que on va vouloir. Donc on va faire des trajectoires, essayer dans notre notre modèle et on va avoir des trajectoires avec plus ou moins de de bons résultats. Ok donc le donc l'intuition de base ça va être en fait, on peut juste faire en sorte d'augmenter la probabilité d'une certaine trajectoire. Sachant que le retour qu'on a sur cette trajectoire est bon. Ouais. Et en fait fondamentalement ce qu'on va faire, donc on va, on va le justifier mathématiquement tout à l'heure. Mais globalement sachant que ce truc là il dépend de péta. Ce serait quand même cool d'avoir de d'avoir un de récupérer les gradients de J pour pouvoir faire une une gradation donc montée de gradient. Bon d'accord ça si j'ai une montée de gradient, bah justement je peux faire ce truc là. Je vais tout le temps dans la direction où je descends, où je descends le plus donc je vous montre le plus dans le cas de de l'ascension. Et je trouve des des atomes locaux comme ça, OK ? Donc voilà, donc si j'arrive à avoir les gradients de J, et Bah c'est facile. Il suffit de dire que thêta prime, il vaut thêta Alpha sur les gradients de J et et c'est fini quoi. On récupère, on trouve le petit moment comme ça. Bon, le problème c'est que. Globalement. Avant la probabilité d'une trajectoire et avoir son gradient, c'est quand même pas évident. Ok, donc globalement en fait de manière générale on a pas ce gradient là. Et donc c'est pour ça qu'on va avoir des soucis et. Et c'est et c'est pour ça qu'on va voir qu'il y a un théorème qui nous permet de calculer le gradient. Dans certains cas il a beaucoup OK sauf que. Bon, globalement ce qu'il faut retenir c'est que de manière générale le Gradient Lee en fait il existe pas, on sait pas le calculer. Vous avez des des environnements qui sont trop compliqués, qui sont pas différenciables. Et donc le gradient vous pouvez pas le calculer. Donc c'est un sujet de recherche aujourd'hui. D'ailleurs il y a des chercheurs aujourd'hui qui travaillent sur. J'ai rencontré un mec, il inria Saclay je crois, et ils essaient de développer un un simulateur de robot, donc un simulateur. Donc dans la vraie vie qui est différentiable, ça veut dire que à chaque moment de votre enfin quand vous faites une action, et cetera, vous pouvez calculer les gradients sur en gros les Gradients du monde quoi, les gradients de quand je bouge mon bras et cetera. Et ça, c'est un travail, mais c'est c'est un truc de fou quoi. Ouais c'est différent parce que là on bah en fait ça s'est basé sur les méthodes qu'on a aujourd'hui et dans l'idée ton environnement en fait, il est tellement compliqué que que ton travail tu as jamais accès. Donc pour l'instant on n'a pas de gradient donc on va parler rapidement des méthodes qui permettent de de faire sans OK. Je sais pas si le terme de métaheuristique vous parle, non ? Donc pour pour pour votre culture, les métaheuristiques c'est toutes les méthodes et en vrai. Enfin je pense que la plupart d'entre vous vont faire de l'optimisation à un moment donné on l'émetteur ristique. C'est toutes les méthodes qui vous permettent de trouver des optimums sans avoir des gradients. Ok, ça veut dire quoi parce qu'en gros. Donc imaginez que ça si votre espace ça c'est votre espace des de J OK donc paramétrisé par tous les États possibles. Et globalement, vous partez du principe, vous avez pas accès aux radios, on fait comment ? Hein ? Et Ben là le globalement le premier truc à faire c'est c'est du Random quoi. On on sample un thêta quelconque quelque part. Hop, on l'évalue, et donc on a un point sur notre courbe, parce que on peut calculer le le, le, le, la reward, enfin le le reward. Donc si on le fait plein de fois, on va avoir une bonne estimation de notre espérance, de notre reward. Ok, on est capable de le faire. Par contre avoir le gradient, ça aucune, aucune idée. OK, donc la seule chose qu'on peut faire pour le moment. C'est juste du rhum, du rhum, simplement OK, et c'est là. Où interviennent les moteurs rustiques d'accord, les moteurs rustiques, c'est en gros, si vous voulez noter un truc, c'est comment est-ce que je fais pour faire mieux que ça ? Avec le la même information. Est-ce que quelqu'un a une idée ? Par exemple, est-ce que je pourrais pas faire un peu mieux que juste faire du du Random du Random comme ça là je vais tout mon espace là là. Imaginons que j'ai tiré le point, le point bleu qui est ici. Euh. Est-ce que je continue à faire un Random partout ? Est-ce que je fais autrement ? Ouais, t'as idée. C'est ça comme ça. Ouais ça regarde les par exemple les 3 points qui avaient les meilleurs, les meilleurs résultats et ensuite tous les autres, ils étaient retirés au hasard. Mais entre ces 3 points là et en fait à chaque fois qu'on a un petit trou de mon plus petit minimum en fait il prend la place des 3 qui restent fixes et. Attends ouais d'explorer après de David. Ok, ce qu'on est d'accord que l'algorithme vient de de de préciser, ça va mieux marcher, on va converger plus vite que en faisant du Random comme ça. Ouais d'accord. Donc ça, ça se base sur l'hypothèse sous-jacente que notre notre fonction, enfin notre paysage de l'os. Comme on l'appelle, il est un peu lisse quoi. Et donc globalement ça part du principe que bah si on est très bon à un point en particulier, bah quand on va être pas loin de ce point là, normalement on va rester très bon, ce qui est pas du tout évident dans le cas général. Je sais pas si vous avez déjà vu, je vous montre ça des nations, de paysages, de l'os. Vous vous souvenez à l'infini en fait. Du coup ça devient des fractales. Parce que votre en gros, enfin ce que ça veut dire c'est que vous avez beau avoir un super. Une super. Euh. Comment dire, euh. Un super résultat pour un thêta donné, si vous décalez un tout petit peu, et Ben vous avez un résultat tout pourri. Et encore un tout petit peu de super résultats. Donc en fait votre paysage est tellement. Et tellement peu lisse que on peut pas faire ça. Parce que en fait justement les les informations de OK je suis à peu près dans cette zone là en fait elles veulent rien dire parce que si tu décales un tout petit peu tu deviens très mauvais. Mais sur l'hypothèse que on a un espace qui est globalement ressemble à ça. Et Ben en effet, ce genre de méthode, donc le la librairie des lots comme tu l'appelles, c'est une métaheuristique, parce que heuristique ça veut dire heuristique, c'est tout, c'est tout ce qui est rapport à un peu. Voilà, c'est heuristique quoi. Mais globalement ça fait je sais pas si le terme heuristique vous parle, mais c'est toutes les méthodes que on va implémenter, ou on sait que intuitivement ça devrait mieux marcher que le que faire du rugby et et ce qu'on c'est un peu de des choses qu'on vérifie de manière empirique. Souvent on va pouvoir trouver des justifications, des justifications. Mathématiques pour dire que quand on fait d'une certaine manière pardon, ça marche mieux, mais. Du coup, je me suis perdu, je sais plus ce que je disais. Euh. Ouais. Ouais ou globalement les. Vois ça donc l'idée du metteur rustique, c'est que vous allez pouvoir. Vérifier manière empirique que ça marche mieux que Random et souvent on va aussi pouvoir trouver des justifications. Que à la limite, ça converge vers l'optimum, même si on ne fait pas quelque chose de de calcul par ingrédient, typiquement. Ok, donc pour l'instant on est sur une route de Search, donc on voit que c'est quand même pas fou hein ? Parce que globalement ça ça ressemble à ça hein. Si on essaie plein de thêtas juste bah on a des trucs Random quoi. Et donc l'idée en effet c'est que si j'y ai à peu près lisse, bah on va pouvoir essayer de faire des. Des choses un peu mieux quoi. Et donc là, le métal rustique. Gardez à l'esprit que c'est valable sous l'hypothèse de. Donc l'hypothèse de la localité, ça veut dire. Si vous êtes très bon un endroit, vous êtes très bon normalement, un peu à côté. Euh pour la culture. Et enfin je pense que c'est vraiment des choses qui sont à connaître. Enfin c'est pas c'est pas à partie du cours du RL mais. Des algorithmes génétiques, le. Alors comment s'appelle le simultanée anning en français, c'est quoi ? C'est le rugby simulé, ça vous parle ? Bah ça, franchement, ça va connaître et. Et les algorithmes de Coline fourmi, je sais pas si vous avez déjà entendu parler de ça aussi, non ? Il y a des supers animations sur Internet. Sur les algorithmes de phonie, c'est très Lego, donc globalement des algorithmes qui nous permettent de. Faire des tirages plutôt dans la zone où ça marche bien et donc converger, OK. Ok. Et donc globalement. Ce qu'on aimerait c'est quand même avoir le Gradient hein, ça ira ça comme ça irait quand même beaucoup plus vite plutôt que de faire des tirages. Mais de manière générale, le gradient de J, on le connaît pas. Ok. Donc là on a vu comment est-ce qu'on faisait dans le cas où on n'avait pas accès au gradient. Et maintenant, on va voir. On va voir un théorème qui nous permet de calculer ces gradients dans un cas particulier. Ok. Bon je vais pas vous détailler toute la preuve mais donc dans l'idée. On part de l'écriture que je vous avais mis en haut, d'accord, on écrit, j'ai comme l'espérance, mais qu'on paramétrique, donc on consomme sur les trajectoires en entier, OK ? On passe au gradient, OK. On multiplie par un. Du coup là j'ai gradient de P sur p, ça vous reconnaissez un gradient du log ? Ouais. Hop radiant, bloc de P. Parce que la dérivée de log de U c'est un sur U, on est d'accord ? Donc quand j'écris gradient de P sur p hop, je peux écrire gradient de log de P. Euh. On a toujours la probabilité sur les trajectoires et donc en fait je peux le réécrire comme l'expérience. Donc on est passé. On est passé de l'espérance directement de J à l'espérance de log de P fois. Enfin l'offre de la trajectoire par le reste. Un bon. On n'est pas hyper avancé quand même. Et mais en avance. Ok. Le problème il est là, on n'a pas d'expression analytique pour P et c'est ça qui globalement nous empêche de calculer le gradient. Ok. Euh. Mais on va voir que. Si on le réécrit avec la justement avec l'hypothèse de markoff, on va pouvoir avoir des choses qui qui vont bien se passer parce que vous vous rappelez que on a réussi à mettre un log et on sait que le log quand il y a un produit ça se passe très bien. Ouais. Donc, et c'est pour ça qu'on a voulu faire apparaître un log très bien. Donc en fait, on réécrit la probabilité de thêta sachant thêta, comme le produit des probabilités de St +1 sachant STAT. Et là encore une fois, on se base sur la probabilité de Markov. Vous vous rappelez ? De manière générale, la probabilité de la trajectoire, elle dépend de tous les États en même temps. Sauf que dans le cas plus simple, on se met sous l'hypothèse de Markov, donc avec des donc avec une chaîne de Markov. Eh bien on va pouvoir séparer ce cette probabilités là en le produit de toutes les toutes les trajectoires possibles qu'on multiplie par la politique, OK. Donc là déjà on passe d'une proximité, d'une trajectoire qui est quand même un peu floue et qu'on maîtrise pas trop quoi à des probabilités de transition qu'on connaît un peu mieux. Et notre politique c'est la politique, on la connaît pour le coup. Et même la politique j'ai envie de dire, on pourrait presque calculer gradients de cette politique vu que si notre politique c'est un réseau de neurones. Quelques radios d'un réseau de neurones y a plus. Juste à faire toucher pas mal quoi. C'est la petite actuelle, ouais, exactement. Ouais bah de toute façon la, la trajectoire, la probabilité de la trajectoire, sachant État c'est bien en considérant Thêta, c'est le le thêta c'est les paramètres de la politique. Donc donc c'est bien, c'est bien ça qui a, de manière sous-jacente dans la. Donc bien se rappeler que dans on se casse, on se casse. Pardon on se place encore une fois dans le cas de la de la théorie de dans des chaînes. Ok, donc. Si on reprend. Ce fameux gradient de log de plus qu'on voulait calculer, et en fait hop on peut remplacer par ces trucs-là. On fait la somme parce qu'on est sur des log. Ok on là on est content parce que en fait, et là c'est vraiment le cœur de le cœur de de ce qu'on vient de voir. En fait ce truc là, il dépend pas de thêta. On est d'accord en fait c'est juste une probabilité de transition ça. Donc quand on calcule les gradients par rapport à tout ça dégage. Et donc là grâce à donc grâce à 2 choses, grâce à Markov et grâce au fait qu'on a écrit sous forme d'un log. Et Ben en fait on fait disparaître ce truc là. Donc on se retrouve juste avec les gradients de log de de Log de la police. Du coup ça supprime. Bah là on calcule un gradient par rapport à Thêta, gradient par rapport à quelque chose de quelque chose qui dépend. Pas de thêta 0 constant donc ça c'est une constante de thêta. Donc ça dégage, il ne reste plus que le gradient de Log de PI. Hop donc en fait là on se rend compte que la gradient de notre de notre trajectoire. En fait, ça s'écrit uniquement sous la forme. Du gradient de notre policier. Et là c'est génial parce que, en fait, notre policier, on la connaît et notre policier, on peut la la différencier. Donc en fait, on connaît le gradient de. Monsieur, juste la notation pour la notation, ce que vous avez mis entre crochet, ce qui sentait c'est ça. Je ça non ? Ouais ouais, ça veut dire que on prend le gradient de tout ça. Donc par somme, ça veut dire pour l'occasion de simple le gradient de l'égalité du. Hop. Donc en utilisant ce qu'on vient de voir. Ou bien de quel sujet-là, hein, d'accord ? Eh Ben on peut réécrire le gradient de J comme l'espérance. De la somme de tous nos gradients. Euh sur la politique. Directement ? Et donc ensuite on peut l'approximer. C'est encore une fois l'espérance, on la connaîtra jamais. On peut l'approximer sur une moyenne de plein de trajectoires possibles. De de nos gradients avec tous les États, donc avec. Qu'on traverse lors d'un épisode OK donc donc là on est super content, vraiment super content parce que en fait tout ça on le connaît quoi. Le Return, Bah ça c'est le l'environnement qui va nous donner. Et les gradients de notre politique qu'on la connaît se réservent qu'on fasse une politique différentielle, mais ça c'est possible. On va le voir et on a vu que on allait pouvoir faire un soft max sur des sur des logs. Donc en fait nos logs ont notre préférence, on va pouvoir le calculer qui a un neurone, et on va pouvoir calculer nos radios. Donc là on est super contents, on connaît, on se connaît pied donc on va connaître. On va connaître l'autre depuis enfin une raison sur l'autre outil. Donc là on on se dit super on va pouvoir on va pouvoir calculer des gradients chirurgie et donc faire une faire inefficace et une une montée gradient beaucoup plus efficace. Donc ça c'est la la formule finale à connaître. Ok qu'on on peut écrire. Donc là il y a 2 hypothèses pour pour écrire ça, c'est quoi le 2 hypothèses ? Ouais, markoff j'ai entendu. Hello, comment ? Hello, ouais enfin les logs, c'est c'est la méthode, mais il faut qu'on, il faut qu'on puisse différencier ça. Ok. Et donc là. Le cœur de l'intérêt de cette formule là, c'est que on n'a pas besoin de connaître l'environnement. Il ne suffit d'avoir. Le Return OK il nous suffit d'avoir la valeur de la récompense à la fin et là je peux calculer un gradient sur l'optimum de ce truc là et c'est assez. C'est vraiment très fort comme résultat. Comment on peut avoir en fonction de l'environnement dans ce qu'on connaît pas, qu'on connaît pas ? Tu vas faire un épisode, c'est le principe du R. Tu vas essayer. Tu prends ton agent avec la policy PI de Thêta. Je fais 100 épisodes. Pour chaque épisode, tu récupères 1RT. Et donc enfin à la fin, tu fais ça, tu fais, tu calcules la moyenne sur toutes tes sur tous tes essais. 2 donc la somme pour toutes et pour tous tes top step de ton gradient. Enfin pas ton gradient du coup, mais quand tu passes au gradient de la politique pour chaque action je sais qu'à ce temps-là j'ai pris l'action AI sachant que j'étais invité à St. Multiplié par la revente que j'ai eu à ce moment-là. Enfin le mérite ça veut. On part du principe qu'on a toutes les rewards associés à toutes les actions qui sont dans notre environnement. Ouais Bah ça c'est le principe du R, c'est à chaque time step ton environnement et te renvoie une récompense et et ton et ton action. Élise, enfin c'est ça que j'arrive pas trop à me rendre compte. Par exemple si on veut faire un entraînement sur un robot. Ouais, ça veut dire que si on veut qu'il évolue dans un environnement précis, c'est d'établir un environnement dans lequel il vous dit et d'associer des récompenses à chacune des actions. Ouais c'est ça. Ou alors c'est l'environnement qui va te donner la convention. Dans le cas d'un jeu, bah la récompense ça va être ton score, ça va être. Après en effet, c'est ce qu'on a vu au début, c'est un peu à toi de choisir c'est quoi ce que tu veux mettre comme ce qui est bon ou pas pour le modèle qu'on souhaite apprendre exactement dans l'environnement ? Exactement. Mais une fois que t'as une récompense, un environnement et des trajectoires, donc une manière d'interagir, c'est-à-dire des États et des actions que tu peux faire pour évoluer, et Ben. Tu peux calculer le gradient de ton de ta fonction utilité. Qui s'écrit comme d'espérance du du retour et juste ce qu'il peut y avoir plusieurs types de récompenses dans un au sein d'un même environnement. Type de récompense, c'est à dire par exemple dans un j'imagine un toujours pour rester sur même exemple. Mais dans les jeux vidéos il peut y avoir plusieurs actions et plusieurs choses qu'on pourrait potentiellement avoir envie que le modèle fasse. Je prends par exemple l'exemple de fortnite, celui qu'on a vu au dernier coup. Bah je sais pas. Par exemple ramasser des armes, tirer sur des gars, ça peut être une récompense mais aussi gagner la partie. Mon dernier, ça peut aussi être une récompense tout à fait. Et donc comment est-ce qu'on ? Enfin c'est ça. Ma question, c'est, est-ce que on peut récompenser de manière différente ? Ouais bah en général du coup dans ce genre de cas tu vas écrire tes récompenses comme la somme de toutes ces petites récompenses. Je sais pas si vous vous rappelez le film que je vous avais mis enfin la vidéo là avec le robot qui apprenait à marcher ? Et bah on au fur et à mesure de l'apprentissage on changeait même la récompense pour d'abord lui apprendre à marcher. Une fois qu'il avait convergé vers cet État là bah on lui apprenait à je sais pas à éviter. Il y avait, on y lançait des cubes dessus ou lança un bah là tu vois c'est tout. Justement, c'est tout le rôle de du de l'ingénieur quand il fait, quand on fait entraîner un robot, c'est ça, c'est choisir la bonne récompense qui va faire que le modèle va avancer dans la bonne direction et va prendre ce. Donc dans fortnite en effet tu vas pouvoir mettre des récompenses intermédiaires du style tu fais pas +1 si tu ramasses une arme, si tu dis +10 si tu tu butes quelqu'un. Enfin tu vois ce que tu veux et +1000 à la fin. Si tu as comme ça le modèle il peut se concentrer à la fois sur OK c'est cool de d'avoir une petite récompense maintenant, mais t'oublies pas le fait que à la fin je veux quand même gagner la partie. Et c'est ça qui est plus important. Ok, donc globalement la réponse à ta question c'est, on fait la somme et on fait une somme en général pondérée en fonction de ce qu'on veut faire apprendre. Donc là c'est justement tout, tout votre rôle, et c'est ce que enfin ce que vous allez faire plus tard dans le cours aussi c'est bah comment ce que je fais pour pour ça ça s'appelle du Reward Engineering. Enfin vraiment ça c'est tout un nom, c'est un peu comme le prompt Engineering avec les LM quoi, c'est qu'est-ce que qu'est-ce que je dis au modèle ? Enfin comment est-ce que je dis au modèle ce qui est bien pour lui ? Ok. Alors, on en est où là ? Tac, TAC, TAC. Euh. 54. Je vais finir ça OK du coup ? En fait, on a tout résolu quoi. Là on a un gradient sur le sur la fonction édité. Et bah il suffit de calculer ce gradient et on fait une on fait une, on a une montée de gradient donc on a on a un algorithme. Du coup j'appelle naïf parce que c'est vraiment l'application bête de ce qu'on a à faire. OK on a notre nos, elle est là ça ça, lorsque j'écrivais. Je suis capable de calculer le gradient maintenant grâce à notre super théorème et je mets ça dans Adam et puis je fais, je fais Los point bac propre et puis et puis je fais la dam. Step et puis c'est bon quoi. Ça a l'air un peu trop simple. Mais pourtant ça marche franchement. Là on a, on a notre trajectoire, on peut calculer, on peut calculer nos récompenses, enfin on récupère nos récompenses, on peut calculer. Mais bon, c'est un peu trop simple quand même. La, des gros problèmes avec ce qu'on ce qu'on vient d'écrire en fait. Euh là, ce qu'on écrit, c'est un peu une estimation de Monte Carlo, on le fait, on le fait, on. On le fait une fois qu'on a terminé l'épisode. Et vous vous rappelez que Monte-Carlo c'était en effet pas de désirer, mais sur ça serait d'une très grande variance. Euh en fait ouais, la somme des récompenses elle est pas, elle est pas tant informative que ça, si par exemple. Si, par exemple. R, il vous 0. Mais en fait vous allez rien apprendre parce que votre gradient il sera nul. Et en fait, avoir un épisode où la reward c'est 0 à la fin, Ben c'est pas impossible. Et même imaginez ou imaginez un cas encore pire ou la meilleure récompense que vous puissiez avoir, c'est 0 parce que votre récompense est négative d'habitude. Donc là en fait il y a juste jamais pouvoir calculer le gradient parce que vous aurez toujours avoir réseau donc votre modèle il va jamais apprendre. C'est un peu \*\*\*. Ouais, on peut faire la loi plus une valeur pour avoir toujours des valeurs. Mais c'est très très juste comme remarque, tu vas voir qu'on va, on va faire ce genre de choses. Donc globalement ce qui a à retenir c'est que l'algorithme naïf ça marche sur sur la théorie. En pratique dans la plupart des cas où vous allez être confrontés, bah ça marche pas suffisamment voir ça marche pas du tout. Parce que oui, par exemple, votre votre gain moyen, il va être nul sur la plupart des des méthodes que vous avez testé et donc maintenant ce qu'on va voir dans la suite c'est. Comment est-ce qu'on fait pour à partir de ce résultat là qui est quand même très fort ? On n'a pas envie de l'abandonner même si ça marche pas sur sur le cas. Et bah on peut améliorer les performances, donc en convergence, en exploration, en stabilité. Donc justement il y a cette histoire de on se rend compte que ouais là-dedans là en j'aimerais bien. J'aimerais bien un peu de normaliser quoi ? Notre notre récompense les j'ai comme il peut avoir n'importe quelle valeur. Y a un truc qui nous dit que en fait je veux pas utiliser la réponse directement mais peut être une notion de Fécamp. Relative à une politique que j'ai déjà par exemple. Donc typiquement, on va en parler de ça. Il y a aussi le fait que. Là on est sur un truc type Monte-Carlo, qu'est ce qu'on avait, ce qu'on faisait à côté de Monte-Carlo pour justement là on disait compter Carlo, il avait une grande variance, il était pas biaisé. Et à côté on faisait quoi à côté de monte Carlo ? Bon, tout va. Comment ? J'ai pas compris la question à côté. Donc on dans le cours précédent, on a mis en contraste Monte-Carlo avec une autre méthode. On disait que Monte-Carlo on a une biaisé parce que on prend le on prend le gain total à la fin. Donc on a une vraie valeur entre guillemets. Ok. Mais le problème c'est que on a une seule valeur quoi, et que bon, elle dépend de, elle dépend de la trajectoire. Et on et on utilise une trajectoire entière alors qu'on pourrait utiliser des. Peut-être moins d'informations. Et du coup, on l'avait mis en contraste avec une autre méthode. Ouais exactement, c'est déjà ça vous parle, je vous rappeler. Ou t'aider learning. Au lieu d'apprendre sur tout l'épisode on apprend vraiment sur toutes les transitions. Donc ça c'est notre épisode. Il fait il y a 1000 Time States dedans. Moi je passe de un sample à 1000 Samples parce que en fait je prends chaque différence par contre. Donc on on sait que le temporel de France c'est moins variant. Par contre c'est beaucoup plus biaisé. OK, donc gardez ça à l'esprit si dans temporal différence vous utilisez une police qui est biaisée parce que voilà dans une direction bah vous allez avoir des approximations dans votre return qui seront très mauvaises parce qu'en fait ça se trouve vous êtes sur sur un chemin où vous allez. Enfin je sais pas, dans fortnite y a y a plein d'armes par terre donc vous allez récupérer plein de choses. Mais en fait à la fin vous allez tomber dans un piège et du coup vous allez perdre quoi ? Et donc votre votre, votre Teddy learning là il va vous dire Ah bah c'est super ce que tu fais alors qu'en fait à la fin vous allez perdre et ça vous le savez pas encore puisque vous faites-vous apprenez sur les transitions. Ok donc là où le Monte Carlo vous aurez dit Ah bah en fait la fin de ton épisode elle est nulle donc ça c'était pas bien comme action. Votre learning ? Lui il va. Il va dire que c'était plutôt des bonnes actions. Donc encore une fois il y a toujours des histoires de de Trade off entre entre tout ça. Ok donc premier tricks qu'on va qu'on va rajouter sur sur notre théorème de de de politique gradients, c'est en effet le le fait de focus en Next World. Ok donc c'est ce que je dis, on passe de Monte-Carlo Easy learning d'une certaine manière ou qu'on va séparer notre reward ? Aussi, enfin notre interne. Pardon donc ici en la somme de toutes les regards jusqu'ici, vous me suivez ? OK. Ensuite, comme les récompenses passées dépendent plus une attraction, alors quand on fait le gradient, on va pouvoir virer, on passe de cas égal à égal T on est d'accord ? Parce qu'on est passé, parce qu'on utilisait des c'est ça juste pas. C'est juste une question de je suis en train de calculer 111 non je parle non pardon je suis en train de calculer un gradient TAC TAC TAC c'est une pardon c'est ça ça ? Quand je suis à t égal t. Ok que je que je calcule le gradient de ces trucs là. Bah en fait les reward précédent elles elles en dépendent pas puisque ça ça dépend des actions que je prends. Du coup je peux les enlever. OK ça va juste les maths c'est pas c'est pas le plus intéressant. Mais dans l'idée, on le réécrit en faisant du du TD Learning au lieu de le faire en montecarlo directement OK. Euh et à la fin on fait pareil, juste on utilise la somme incomplète ? On utilise cette somme là. Plutôt que d'utiliser la somme, la somme alors ? Donc ça revient à faire vraiment du c'est vraiment équivalent à la ce qu'on faisait était des learning à ce monde car dans 1/4 on utilise le le retour en entier dans l'autre en fait on fait la moyenne sur les transitions vraiment en tant que tel. Donc là on a moins de variance parce que on ignore les les rewards qui sont pas intéressantes, donc certaine manière que on fait des transitions on a plus d'informations. Ok 2e, 2e improvement, enfin pardon amélioration. On fait du Reward Discounting, on utilise de cdiscount Factor que vous connaissez bien normalement. Donc là c'est la même idée, c'est la même intuition sur cdiscount Factor qu'on avait déjà développé dans les cours précédents, l'idée que on veut. On veut prendre en compte la reward dans 1000 time steps. C'est quand même bien d'avoir une récompense à time step +1 code OK, parce que si on survit pas à l'instant T, on pourra jamais gagner à la fin. Ok, donc ça en pratique ça, ça améliore la stabilité de votre entraînement parce que. Quand vous êtes, quand vous modifiez un tout petit peu la policy, vous, votre, votre, votre. Appartement bah si vous modifiez un tout petit peu votre policier mais que du coup votre comportement change sur les 1000 prochains tests. En fait la variation sur le la récompense que vous allez avoir dans 1000 Timestamp est énorme, alors que la variation qu'on a sur le timestamp +1 elle est plus petite que peut-être on va marcher un peu plus à droite que qu'on avait prévu donc on change un petit peu. Par contre dans 1000 Times bah on se fait enfin si on reste sur Town on se fait buter alors fois ce serait pas fait. C'était ? Donc dans le cas dans dans ce cas-là modifier un tout petit peu notre. D'autres policiers ? Ça a une très grande variance sur les compétences et donc une très grande variance sur notre vie. Et donc on a un entraînement qui va être pas très stable quoi. Donc le fait de de de rajouter ce truc là hein ? Et Ben en pratique ça ça ça améliore la convergence. Ok, et donc là ça nous mène sur. Notre premier algo. Donc modèle Free, donc un premier algo vraiment qu'on peut utiliser dans la vraie vie pour. On peut utiliser dans la vraie vie pour pour faire du vrai. Et c'est un halo qui s'appelle reinforce. Donc là, pendant le, le nom est à connaître après le. Si vous pouvez retenir, c'est que R force, c'est Monte Carlo Pricilla. Tac TAC, donc on a écrit. On écrit pareil notre notre gradient comme on l'a écrit précédemment. Simplement la, je l'ai écrit, on peut l'écrire avec plus si on veut ou avec le gain directement. Euh. Et on fait notre on fait notre date comme ça, OK ? Et donc là on est en mode Monte Carlo donc c'est-à-dire qu'on n'utilise pas la, on n'utilise pas la temporelle différence comme on comme on vient de voir avant. Ok. Donc c'est pour ça que on va avoir une on va avoir une grosse variance sur cet algo là mais ça peut marcher dans certains cas où typiquement on va avoir des épisodes pas trop longs et donc notre estimation de Monte Carlo va pas être trop dégueu. Donc là vous avez le. Ça c'est implémenté, donc. Vous pouvez l'utiliser tel quel, quoi ? Donc là ce que vous pouvez voir, c'est que maintenant vous comprenez ce qu'on fait. On utilise bien le cdiscount Factor qu'on vient d'utiliser parce que on se rend compte que ça de meilleure qualité et parce que c'est plus stable. Et c'est aussi un peu pour ça, mais parce que c'est plus stable. Et Ben on peut se permettre de le faire. Enfin on réduit un peu la variance qui est inhérente au fait qu'on utilise du Monte-Carlo. Je sais pas si si ça vous parle. Et du coup on fait notre bête comme ça on calcule notre gradient grâce aux politiques gradients avec notre gain qu'on a qu'on a estimé avec sur. Ouais. Donc là vous rendez compte que on fait bien, on a un épisode en entier. Mais au fait, un. On fait une estimation de notre gain pour chaque time step de l'épisode. Ouais. Donc on fait pas une seule update pour un seul épisode ? Parce que, en fait, à chaque timestamp dans notre dans notre trajectoire, on a l'estimation du gain. En fait, il suffit de la tronquer si vous voulez. Si on est à le gain, c'est l'espérance de. Le gain, c'est quoi ? L'espérance de. Ouais. Et quelle reward du coup ? Ouais, à partir de quitter. Et du coup ici t'es égal t. C'est le, le pardon, le return. Ouais, c'est pas, c'est pas un. Avec le grand thé en général, on existe pour. Donc là, c'est. I égal. Ok. OK donc ça ça veut dire que pour chaque thème step dans notre trajectoire en fait le gain je je l'ai donc simplement si j'ai toute une trajectoire là s 0 à 0 un jusqu'à gantier ? Bah en fait là avec on. Quand je prends toute la somme de R de R un à RT, j'ai j'ai un. Si je prends la somme à partir de R 2, je vais estimer j'ai 2. J'ai 3, j'ai 4 j'ai 5 externes donc c'est comme ça que je vais avoir t estimation de G pour chaque aspect et donc pour chaque pour chaque reward que j'obtiens je peux faire un update de ma police donc ça ça me permet typiquement de. Si jamais votre modèle. Il a il a fait une très mauvaise action au début de la trajectoire. Bah typiquement y avoir des très mauvaises rives donc y a 0 et un qui sont très mauvais par exemple. Mais comme j'ai gamma en fait ça va il va se concentrer sur sérieux là. Donc je vais avoir un return qui va être négatif donc on va punir le modèle pour les mauvaises actions qu'il a pris au début. Donc on va avec la gradient, on va réduire la probabilité qu'il refasse cette même action plus tard, OK ? Par contre, imaginez qu'ensuite il a fait des très bonnes actions et qu'il a fini par gagner la partie. Bah ça veut dire que. Quand je vais avancer dans mes dans mes tests pour faire mon update bah je vais oublier les premières rewards que j'ai qui ont été très mauvaises donc quand je vais calculer le G 10 par exemple. Eh Ben il y aura plus ces mauvaises récompenses dans dans le gris, donc je vais pouvoir apprendre. Enfin, le modèle va pouvoir apprendre à à refaire, à renforcer ces bonnes actions qu'il a fait. Là-dessus. Ok, donc ça c'est rien de force, ouais. Il suppose que la somme de toutes les récompenses, elle soit égale à un à la fin. En gros, c'est une sorte de pourquoi ce qui serait égal à un. OK Bah j'ai mal j'ai mal compris alors pourquoi ça serait y a un là ça me récompense c'était pour l'estimation des gains ce que vous dites enfin ce que j'ai envie de vous dire c'était que pour pouvoir essayer un gain par exemple si vous y mettez un bah on fait la somme de tous. Ben de toutes les eaux de recouvrance, moins comme on avait le le fait que ça c'est un, on pouvait trouver le. On va pouvoir, on pouvait estimer le gain au moment mais c'est pas ça que vous avez du coup. Et c'est ça que j'ai compris. Non ? Bah du coup non c'est ça. Ben je vais, je vais réécrire ce donc là si on fait une estimation. Très très brutal. C'est à dire qu'on a un seul sample. Là ça veut, ça veut gamma. Juste. Par 2 plus. Ils ont qu'en gros. Là, je vais avoir. Eh bien. Ok. Ensuite ici je vais avoir karma R 2 enfin gamma G 2 pardon là je vais voir gamma carré G 3OK ça c'est juste pour vous dire que si j'ai. Une trajectoire en entier. Demoiselle 0 jusqu'à testé là et là je peux estimer. Un retour, enfin un gain ? Pas et donc je peux. Donc c'est là, on combine 2 idées, enfin 2 tricks. Là donc dont on a parlé, on utilise à la fois tout, toutes les transitions qu'on a connues. OK. Euh. Et à la fois, on utilise le gamma pour comment dire, normaliser ou plutôt. Renforcer la stabilité de notre. Ok. C'est bon pour une force. Ok, on va faire une pause ce moment-là. Ouais, euh. On va faire une pause maintenant je pense, donc ça fait. Là il reste en gros 30 45 Min de cours. Je vais encore 2 albums vous montrer et le TP va être un peu plus long donc je pense c'est bien si on fait une pause maintenant et. Vous avez des questions sur une force si vous voulez qu'on vienne à quelle heure ? Bah du coup 15 Min à 09h30 Vous êtes là, ça marche. Merci d'avoir. Beaucoup ? L'interroger. Ah si on part de 0. Euh, ça c'est les TRT ? Ouais, c'était égal 0, on on a juste 0 quoi. Ouais, ouais, j'ai. Par exemple, dans. C'est moi la 2nde ou pas ? Le RP, c'est quoi ? J'ai un problème avec. C'est bien le départ. Je suis dans un État, mais je reçois une. En général je reçois ton intervention mais c'est pas obligé, j'en ai robot tu vois mais. En particulier. Ça ? Les trucs ? Pas besoin. J'ai pas besoin de l'action de de. J'ai pas besoin de sinon ce serait ça ça. Du coup le. Après. Tout le monde estimateur à. Donc maintenant on va passer sur des méthodes, sur des méthodes dites. Acteur critique, mais on va comprendre pourquoi. Et donc pour ça, on va en voir 2. À tousser et à 3C. Ce sont 2 algos là on commence vraiment à arriver sur des algos qui marchent vraiment bien et qui sont pas mal utilisés dans la vraie vie. Et donc. On va parler d'un nouveau nouvel astuce qui est précisément celle dont on parlait avec l'idée que. Bah, la récompense en elle-même. Ben on n'est pas hyper content. Parce que si jamais elle est nulle ou si jamais elle est négative, Eh Ben on va avoir des problèmes. Donc par exemple donc là si on écrit notre grade en comme comme l'a écrit juste ici. Euh. Eh Ben, imaginons que toutes les revues sont positives, ça fait possible. Eh Ben si toutes les revues sont positives, Eh Ben. Le gradient, là il va être positif partout, donc il va tout le temps. Enfin il va. Pardon négatif partout parce que ici on a un truc négatif. Il va augmenter toutes les probabilités. Ça veut dire que je vais pas être capable de dire que le modèle devrait diminuer la la la propriété donc le log qui va être. Ok mais c'est quand même bizarre parce que ça veut dire que si je rajoute une une constante dans dans dans mes rewards bah ça va changer mon ça va changer. Enfin comment dire. Enfin ça devrait pas dépendre de ça quoi. On aimerait bien que. On aimerait bien que ce soit vraiment une notion de c'est mieux de faire cette action là peut-être plutôt qu'une autre ou dans cette situation là quoi ? Euh. Donc il y a un peu l'idée de on va vouloir un peu renormaliser ce truc là pour que ça aille plus de sens. Et donc on peut montrer que, en fait, je peux ajouter une ce qu'on appelle une baseline, donc une constante ici. Sans que sans que ça change la moyenne. OK donc pourquoi est-ce que faire ça ça marche ? C'est parce qu'il y a une moyenne. De manière générale c'est je peux pas juste rajouter ça à l'intérieur de de récompenses. Par contre ça marche parce que on est en train de calculer une valeur, une valeur moyenne sur plusieurs épisodes. Ok. Donc mathématiquement, ça c'est correct. Euh. Et. Et donc on va pouvoir faire donc précisément ce dont tu parlais un peu l'idée de on va enlever une valeur pour que pour que le modèle puisse. Prendre par rapport à une baseline plutôt que par rapport à rien. Et donc dans le cas où on a des retours qui sont nuls Bah voir des Gradients et pas pouvoir apprendre. Donc par exemple. Une une première maison qu'on pourrait voir, ce serait le le le return, donc enfin le le return moyen. Sur toutes les sur tous les sur notre badge. Donc là ça veut dire que si j'enlève si mes moins herbart ici. Normalement, enfin, la récompense moyenne. Ça veut dire que ici, ça veut devenir positif quand la récompense est mieux que d'habitude et négatif moins que d'habitude. Là c'est pas mal. On se dit Bah ça veut dire que le Gradient va être positif quand la concurrence est meilleure que ce que je prends d'habitude et négatif quand c'est moins bien. Déjà on peut dire ça, ce sera un peu plus logique que juste mettre la récompense comme ça. Ouais. Au niveau de tout à fait. Donc on récupère une moyenne de 0 et on peut même diviser par la par la par la Variance pour avoir enfin non pardon l'écart type pour avoir un écart type de un sur sur apparence. Donc là c'est pareil, une question de stabilité comme ça on normalise. On normalise et on est sûr qu'on a des meilleurs. Un meilleur comportement de général au niveau numérique en particulier, donc. Donc on peut arriver à renormaliser tout ça. Hop. Ok et donc là on a un nouvel algorithme certaine manière on va avoir un policier gradient avec une base constante qu'on va écrire. On va faire comme l'algorithme précédent, sauf que cette fois, on va écrire la formule avec R +1, enfin avec DRT, moins la. Moi, la valeur moyenne ? Et en effet là on a un petit, on a un petit problème, c'est que c'est ce que tu disais. On empêche pas le fait que ne pas avoir 0 quoi c'est quand on a 0 ? C'est donc ça, ça peut marcher dans un cas où on on a une récompense qui est vraiment différente à chaque épisode. Typiquement dans le labyrinthe ça va pas marcher. On a la même récompense tout le temps donc on va pas pouvoir faire grand chose. Mais si vous avez un un jeu où vraiment vous pouvez avoir une bonne récompense, plus une mauvaise, et cetera, là ça c'est un truc qui est une étape qui peut fonctionner. Hop on. Toc. Une autre blague qu'on peut utiliser, ce serait la Steve fonction pourquoi ? Pourquoi c'est une bonne idée d'utiliser la State value function ? Là on a dit si jamais on a des récompenses, par exemple dans le cas du labyrinthe. On a vu qu'en fait les récompenses à thé elles sont un peu, elles sont un peu inutiles, on a tout le temps 0, on a juste un à la fin donc pourquoi est ce que c'est intéressant de passer à la value function plutôt ? Ouais. Ouais ça ça c'est dans le c'est c'était le cas de la value itération mais. Mais du coup, quand on faisait cette opération là, on en finissait avec quoi dans le labyrinthe typiquement ? Au lieu d'avoir des récompenses tout le temps pareil que ce qu'on avait au niveau de la fonction. Exactement. Donc elle était beaucoup plus informative. En fait, ça veut dire que même si c'est mon objectif, c'est de sortir de cette salle. Même si j'ai une récompense de moins un à chaque pas et Ben si j'utilise ça va lui function là j'ai une fonction qui augmente parce que je me dirige ma bonne ça veut dire que si je remplace dans la baie d'ici là pour augmenter mes gradients dans les bonnes actions. Et bah si j'utilise une machine constante Ben en fait j'aurais tout le temps 0 donc rien apprendre hein par contre si je la value function. Et Ben là on voit bien que plus je me rapproche, plus j'ai les meilleurs valus fonctions et donc plus. La vraie différence entre Marie que j'attends à. Temps maintenant c'est nul et par contre une belle fonction qui elle augmente ça veut dire que certes d'une récompense nulle mais en fait ma solution chaîne a augmenté donc on l'action que je viens de prendre elle est bonne. Ok. Ça me donne une meilleure estimation, par contre ça veut dire qu'il faut connaître la value function, ça c'est pas évident. Mais donc l'intuition c'est qu'en fait si on connaît la veille function c'est la meilleure de veillon quoi. Et donc c'est pour ça, on écrit comme ça. On l'écrit comme ça, euh. Et ça s'appelle la fonction avantage OK. Donc l'avantage ça va être donc la la State action value moins la value fonctionne. C'est à dire c'est quoi l'avantage de quand je suis dans cet État là de faire l'action ? Par rapport à la value function que que je suis pas comment dire la value function associée à l'État dans lequel je suis ? Donc c'est quoi l'avantage de prendre ? L'action a par rapport au fait de juste me trouver dans l'état de. Ok, intuitivement c'est vraiment en fait c'est dans quelle direction que je vais augmenter un maximum ma vie fonction ? Ok, donc on définit la fonction avantage comme ça ? Je ouais, y a une question que vous avez posé là pour vous avez dit à quoi ça ? En quoi c'est mieux de prendre l'action a par rapport à l'État ? Euh donc l'idée c'est que c'est qu'on va pouvoir ajouter une baseline dans notre. Dans notre. Dans notre condamné en formule OK, parce que précisément. Si j'ai une baseline de s qui ne dépend pas de de thêta. Notre gradient, il est nul, ça veut dire que. Je peux avoir une baseline qui n'est pas constante mais qui dépend de s. Donc ça veut dire que je peux prendre une. Je peux prendre une balle function par exemple je pourrais pas mettre une cuve function parce que ça dépend de ça dépend de l'action et donc ça dépend. Et cetera, ça changerait mon gradient. Par contre en gros ce que ça veut dire c'est que je peux mettre s à l'intérieur OK ? Et donc je peux mettre pardon V 2 à Internet et donc pourquoi est ce que c'est intéressant ? On a vu parce que ça apporte plus d'informations que une reward. OK parce que la vallée Fonction, elle prend en compte toutes les rewards possibles et donc à partir du moment on la connait. En tout cas on. Donc on se rend compte que si on remplace donc dans notre formule, si on remplace. Par la fonction on peut faire apparaître la cuve, la cuve, l'action Valeo fonction et donc on se on se retrouve en gros à définir cette fonction là. Donc ça c'est une fonction à connaître la fonction avantage OK. Et donc la fonction avantage, c'est la différence entre l'action value function OK et la value function et donc. C'est quoi ça ? Qu'est-ce que c'est ? L'action stupide de l'espace ? C'est quoi ? C'est quoi la définition de cupide ça ? Xbox. Parce que là, on. Ouais, j'ai entendu, espérant du bien. Tester, donc, ça vaut. Hop. Hop moi. Là, il fonctionne qui dépend aussi de tout. Mais uniquement de s 2OK. Et ça, c'est quoi ? Ok. Donc ma fonction avantage en fait elle veut ça. Vous comprenez pourquoi davantage ou pas ? En fait là dans les 2 cas dans le même État je suis dans St par contre là je fais la différence entre le suis dans l'État St et je prends l'action AT et je suis dans État t donc donc en gros. Si. Cette fonction avantage, elle est positive. Ça veut dire quoi ? Si ça fonctionne, avantage, elle est positive. Ça veut dire quoi ? C'est avantageux de faire l'actu ? Exactement. Ça veut dire que. Les mon espérance, elle l'augmente quand je prends l'action, d'accord, donc c'est avantageux de prendre l'action. Ils sont pas bêtes, hein ? Les gens qu'on qu'on a donné le nom, ils ont-ils ont donné un nom qui a du sens, OK. Alors que si l'avantage est négatif, c'est que quand je prends l'action à dans les les stands. Mais en fait, c'est exactement ça qu'on veut quand on fait notre quand quand on fait le calcul de gradient. Si on si on revoit la formule, on écrit avec les familles des trajectoires, là on veut augmenter la probabilité pour les trajectoires qui ont un meilleur retour. D'où le fait qu'on écrive ce truc. Ok, à bien connaître la la fonction avantage et donc. Donc là faut me faire confiance. Mais il y a je vous ai mis un lien avec toutes les tous les calculs, si ça vous intéresse c'est pas très beau à voir mais donc avec avec cette idée-là je vous je vous ai-je vous ai justifié la plupart des des choses hein mais. Euh en gros l'idée c'est qu'on peut rajouter cette constante là et on peut même rajouter une constante de S et donc je peux rajouter la place de B. En gros la value function et donc je fais apparaître ma fonction avantage à l'intérieur de cette formule là et donc je me retrouve finalement avec un gradient de de notre fonction utilité qui s'écrit comme ça OK ? Donc la fonction avantage OK. Euh. Donc ça à bien connaître, c'est ça ce que ce que je vous disais, ça mesure à quel point est-ce que. À quel point est-ce que notre. Le fait de prendre l'action, ça nous mène dans une meilleure situation, une bonne situation. Ok, petit retour sur le. Sur le compromis entre biais et variance. Vous vous rappelez hein ? Entre entre Monte Carlo et étudier learning ou bootstrap, OK. Dans le cas de Monte-Carlo. On prend tout pour qu'on est. Non biaisé, mais. Variance. OK. Ouais bon, c'est un résumé de ce qu'on a déjà beaucoup dit quoi avec Ouais, on rajoute la notion de de bootstrapping dont on a déjà parlé. Mais ça strapping. C'est encore une fois j'ai une trajectoire, je rajoute petit à petit des petits, des petites, des petites transitions. Ouais enfin c'est ce qu'on dit, ce qu'on dit quoi. Et pour pour tempérer un peu l'inconvénient du bootstrapping qui est que on est, on est très biaisé. Bah on peut faire du Hand, step, e-Learning, on va prendre à t'aider. C'est une seule step. Et instead on prend pas toute la trajectoire, on en prend le disant ici avec que donc que 4. Et en plus on utilise un on utilise un lambda lambda return pour pour pour pondérer OK. Donc ça contrôle notre notre, bien notre et là. L'idée que vous voyez votre biais et votre variance induite tous les 2, une erreur. Et donc il faut contrôler le bon compromis entre avoir finalement peu de biais. Finalement peu de variance pour avoir. Un modèle qui apprend bien. Ok, donc maintenant on a tous les. On a tous les éléments pour pouvoir faire du. Pour pouvoir faire du actor critique. Euh. Qui était ? Je vous ai mis un petit un petit résumé TRPO il me semble qu'on a fait on a pas encore fait, je sais plus. Peut-être après, mais en tout cas on part de police ingrédients OK et. On s'oriente vers les les, les politiques qui sont élastiques. Et toc. Toc toc. On va voir pourquoi est-ce qu'on parle de critiques ? Est-ce qu'il y en a qui ont une idée ? Pourquoi ? Parce qu'on a parlé de critique maintenant. Là, on a vu que on va. On a vu que. Hop. On a vu qu'on va s'intéresser à à ce truc là à l'avantage. Donc globalement la value function on la connaît pas. Donc en fait on va l'approximer la fonction vocale. Et en fait, quand on si jamais on écrit ça. La formule en rouge. Fonction avantage. On va avoir un estimateur. Donc en fait on va avoir un estimateur qui va nous dire à quel point est ce que prendre l'action a dans l'état s c'est bien finalement ? Finalement c'est ça la fonction mentale, c'est juste on voit un truc positif si le prendre l'action a dans l'État c'est bien et un truc négatif si c'est pas bien. En fait ça c'est être un critique. Ouais ouais, en fait c'est un modèle qui va critiquer le fait de prendre l'action dans un État. C'est pour ça que quand on quand on fait des méthodes comme ça, on appelle ça des méthodes critiques, parce que notre acteur c'est la policier et notre critique, c'est lui qui va critiquer la politique, c'est à dire Ah, c'est pas bien en fait de faire de faire ce que tu fais. Ok. Ok. Et donc là là. Et donc là on a plusieurs manières de faire, on peut soit choisir. De calculer ma fonction value directement. Et à partir de la fonction value estimée et estimer ma mon avantage, soit. Apprendre directement. Apprendre directement, bah que action validation vous vous rappelez dans dans des QN, c'est un peu, c'est un peu particulier. On donnait bien l'État en entrée, mais ils nous sortaient à chaque fois. Il nous sortait à chaque fois. C'était c'était ce cas-là, on mettait, on mettait l'État en entrée et une sortait à la fois. Enfin pardon nous sortait la valeur pour tous les toutes les actions possibles. Donc ça revenait avoir une table. Mais on l'écrivait un peu différent. Ou alors on peut le faire vraiment comme ça. Et on a vu que ça marchait un peu moins bien, mais on met vraiment action et État pour obtenir la valeur de la de la qualité humaine. Donc à Toucy, ça ça vient de ce que ça veut dire avantage acteur critique. Ok, c'est pour ça qu'il y a 2A, pour ça qu'on dit à tous. Et donc dans ce cas-là on est, on est acteur critique et on police. Comment est-ce qu'on va calculer un avantage ? Eh bah dans atrocity on calcule notre avantage comme ça. Ok. Donc on estime donc la définition. C'est bien ça hein ? On est d'accord ? Mais encore une fois on a, on n'a pas accès à ces valeurs là. Et donc ce qu'on fait, c'est que on l'estime. En calculant. En calculant FT avec RT qui est le qui est l'estimation du du du Return au total du gain OK. Donc c'est. Le return of the end step tractor for the. On estime donc c'est un peu du c'est du bootstrap hein. Vous voyez on on a en step dans notre replay buffer. Et on calcule l'estimation totale de notre de notre bien, notre return. Avec l'approximation qu'on a de la value function, donc dans advantage acteur critique, on modélise. Donc on paramétrique une value function OK et donc on estime notre avantage comme ça. PAF, PAF, PAF et donc. Donc ça revient à dire que ce truc là là. C'est une estimation de notre variable que. C'est logique parce que on on estime un return et c'est bien un return sachant que je prends l'action a ? OK. Ouais. Et donc on met, on est dans dans un dans un compromis entre juste du TD Learning in step et du Monte Carlo parce qu'on prend en stop, OK ? Avec l'idée que c'est un peu le meilleur des 2 mondes justement, on prend la fois plusieurs steps. Donc on a moins de variances moins de biais, mais on prend pas, on prend pas tous les termes entier, donc on a moins de variance. Je ok donc là vous avez compris globalement comment marche adouci. On a écrit la formule vous, vous savez ce que c'est et vous connaissez votre gradient. Parce que on l'a, on a écrit un peu plus haut. Donc là on est capable de le Coder maintenant. Là on a, on a tout défini. Ok, et donc globalement à tous si on on on prend un peu le meilleur de tous les mondes quoi. On on fait à la fois du Monte-Carlo et à la fois du TD Learning. On se met un peu entre les 2 avec un strip tease learning, on récupère un avantage et donc on n'est pas biaisé par la valeur des rewards. On estime, on estime value function, donc on a un critique et on estime notre policy, donc on a un acteur, donc on a du acteur critique. Ok, donc. En résumé ? En résumé ? Ce qu'on pourrait tenir, c'est que si on veut faire du Poly Gradient qui marche. On a besoin de la fonction avantage OK parce qu'on a vu qu'on avait pas la fonction avantage, on a pas notre baseline du coup juste car la police gradient avec les rewards c'est c'est trop peu, trop peu d'informations, ça marche pas OK ? Pour a tousse. On calcule notre avantage comme étant en modélisant notre value function et en estimant notre action value function précisément avec notre value function. Faut qu'on calcule 7D les rewards et on ajoute gamma fois la value fonction de l'État d'après. Ok donc c'est un peu du. C'est un peu comme bellman, mais avec. Ouais. Euh. Ça je l'ai, je l'ai pas détaillé mais en gros on rajoute une une régulation par anthropie pour. Pour l'exploration. Ok donc dans notre softmax en gros pour calculer les policiers à la fin. On va ajouter une température. Vous allez vous ai déjà parlé de ça, donc on va faire en sorte que, au début, typiquement quand le modèle est pas suffisamment appris, on va continuer à exp. Merci beaucoup OK, donc on utilise des updates de un step et pas de Replay Blocker. Donc donc, donc. Pourquoi est ce que pourquoi est ce que on écrit que on peut pas ? On peut pas assimiler acteur critique et on pourra pas trouver de de modèle de d'algorithme acteur critique et policier en même temps ? C'est quoi un c'est quoi un angriff of policy ? Ouais, nous certainement. Et donc pourquoi est ce que quand on est acteur critique on peut pas être autre policier ? En gros, quand on est acteur critique. On modélise directement notre policier justement, et donc on se base sur cette policier pour faire. Du coup ? Si on était un policier, en fait on aurait pas utiliserait pas notre policier pour faire pour se déplacer. Donc en fait on n'aurait pas d'acteurs. On pourrait pas, ce serait juste un critique. Bon, pas vraiment d'acteurs critiques. Ok, finalement on fait acteur Christ, c'est-à-dire un acteur, donc c'est donc ça veut dire qu'on modélise la politique ? C'est pas, c'est pas le plus important, c'est vraiment de la nomenclature. C'est vrai, le. Ok. Euh. En résumé, c'est pas très clair et donc. On considère que l'acteur critique on est sur du on est sur du off du un policier pardon il y a l'histoire du Replay lever que si on l'utilise. Vous rappelez les replay buffer c'est on retient toutes les toutes les transitions. Ça veut dire qu'on a plus besoin de la policier vu qu'on connaît les transitions directement. Donc on peut dire qu'on est of policy quand on utilise un replay buffer, en l'occurrence acteur acteur advantage, acteur critique de utilise pas le replay buffer. Mais c'est pas évident quoi, du coup on va préférer séparer plutôt. Plutôt des méthodes comme étant value base et policy base OK donc là. C'est normal si vous les connaissez pas toutes, mais à Toucy et des QN ça vous les connaissez-vous des qd on le la dernière fois et aussi en advantage acteurs critique on aujourd'hui on va dire qu'ils sont value base parce que on va approximer la value function là où reinforce. Qu'on a vu aujourd'hui aussi. Vous voyez une force ? On modélisait jamais la fonction, une force. On a juste. Si je le remets ? Euh. Avec une force, on modélise jamais avec fonction, on calcule simplement en à un gain avec les les récompenses qu'on a récupérées. Pas d'approximation, on modélise pas la vie toxique, et donc là on va dire qu'on est polythéiste, parce que on on se base vraiment sur la politique et pas sur. Euh. Voilà, et donc une une typiquement une heuristique, c'est que quand vous allez faire des des méthodes qui sont policy base, on va avoir tendance à avoir un meilleur réseau de neurones pour la policy. Dans un plus gros réseau quoi vous avez-vous avez passé plus de temps à optimiser alors que dans le cas de de d'algorithme value base ça va être plus important d'avoir une bonne value function. Ok. Là, c'est une heuristique. C'est une historique. On se rend compte que ça marche à peu près. Ok. Grâce à 01h40 pour faire le TP. C'est pas de trop, donc aujourd'hui on va passer sur. On va implémenter un acteur critique. Sur l'environnement du du pendule, je sais pas si vous connaissez cet environnement. Vous met le lien sur moodle ça vous va ? Non, c'est bon. Ok du coup le vous avez bien accès au PC c'est bon ? Je l'ai mis sur moodle normalement donc. Le but de de l'algo ici là, c'est de faire un un modèle qui arrive à faire tenir un pendule en position haute, OK ? Et le problème est un peu le même que pour le pour le Mountain, car. Y a pas, vous avez pas assez de puissance pour faire monter le truc directement donc il faut utiliser un peu la l'inertie du l'inertiedu.de vue pour le faire monter, OK ? Donc le l'environnement, là on a action continue et état continu. Ok parce qu'on a un angle et on est on met une puissance dans d'un côté ou de l'autre. Et donc pour pour discrétiser nos pour discrétiser nos actions on va encore utiliser tel coding, OK ? Donc on utilise le même. Le même bout de code que il y a 2 séances avec le mental car justement et donc bon après je vous laisse je vous laisser. Je vais vous laisser voir ça tranquillement, c'est bien expliqué et vous hésitez pas si vous avez des questions. Oui. Ouais. C'est complètement. C'est une artistique, donc sait que. Dans tous les cas, il y a pas de solution simple, donc il va falloir explorer toutes les tous les chemins possibles. Simplement va explorer préférentiellement les chemins qui se rapprochent pour. Parce que, en fait, en moyenne, quand on fait ça parce qu'à un moment donné, il faut quand même se rapprocher du but pour aller de à à B, Même si il faut faire. Donc ça c'est une heuristique. Par contre la value function en effet dans le cadre 40 on peut en. Avec la validation on peut calculer mais mais je dirais pas qu'on utilise des. Donc, quel que soit le. Ça veut pas dire que tu vas pas la calculer. L'espérance de gain, c'est pas évident. Estimateur tu sais qu'il va être \*\*\*\*\* ou va \*\*\*\*\* tu les laisses dans à tous, c'est ce qu'on a fait. Donc si on a, on a pris un beurre compromis on donc j'ai anstett. Et un peu variant. Mais en mais. Mais c'est encore une fois par une restriction juste que on perd, on perd un compromis sur les estimateurs, mais ça reste un estimateur, c'est pas. À star. Choisi de mettre en poids un peu quelconque pour la tente de ton à quel point ton modèle va être biaisé vallée. Que là. À la limite ? Moins de variance, c'est un gamma qui est très faible, donc tu vas très vite annuler les. Du code suivant. Mais. Mais tu prends un peu plus que. Ok. Je veux brancher. Ok.