Dota 2 with Large Scale Deep Reinforcement Learning

OpenAI et al., 2019

Présentation d'un article de recherche

Sacha Benarroch Yoann Ayoub Corentin Felten Baptiste Viera

UTC - GI02

IA02, Printemps 2021

Sommaire

Table of Contents

Dota, le jeu

- Jeu vidéo : RPG stratégique en temps réel

Dota, le jeu

- Jeu vidéo : RPG stratégique en temps réel
- Multijoueur en équipe à somme nul

Dota, le jeu

- Jeu vidéo : RPG stratégique en temps réel
- Multijoueur en équipe à somme nul
- 2 équipes de 5 joueurs

Les résultats observés

- OpenAI Five bat les champions du monde en avril 2019

Les résultats observés

- OpenAI Five bat les champions du monde en avril 2019
- Sur 7000 parties en ligne, 99,4% de winrate

Les résultats observés

- OpenAI Five bat les champions du monde en avril 2019
- Sur 7000 parties en ligne, 99,4% de winrate
- Évolution du style de jeu qui dépasse les stratégies humaines

Les difficultés posées

- Difficultés liées au jeu lui-même

Les difficultés posées

- Difficultés liées au jeu lui-même
- Difficulté liée à l'évaluation du niveau de jeu des agents

Les difficultés posées

- Difficultés liées au jeu lui-même
- Difficulté liée à l'évaluation du niveau de jeu des agents
- Difficultés liées à la conception et l'optimisation d'une IA sur une longue période d'apprentissage dans un milieu en constante évolution

Table of Contents

Table of Contents

Un espace d'observation et d'action de grande taille

Le flot d'information fourni par *Dota* est quasiment continu.

Solution :

- Discrétisation de l'information : chaque *frame* porte l'information à l'instant t.
- L'IA agit toutes les 4 *frame* (*timestep*).

$\Longrightarrow \simeq$ 20000 actions/partie.

Un espace d'observation et d'action de grande taille

Environ 16000 variables sont nécessaires pour qualifier un état du jeu (observées à chaque *timestep*).

$\Longrightarrow$ **facteur de branchement** exponentiellement grand.

	Taille d'une observation	Facteur de brcht. moyen
Échecs	1000	35
Go	6000	250
DotA 2	16000	8000-80000

Table: Comparaison des tailles d'observations et des facteurs de branchement

Un espace d'observation et d'action de grande taille

Environ 16000 variables sont nécessaires pour qualifier un état du jeu (observées à chaque *timestep*).

$\Longrightarrow$ **facteur de branchement** exponentiellement grand.

	Taille d'une observation	Facteur de brcht. moyen
Échecs	1000	35
Go	6000	250
DotA 2	16000	8000-80000

Table: Comparaison des tailles d'observations et des facteurs de branchement

Solutions :

- Quelques simplifications apportées : limitation de la prise en charge des héros et des *item* spéciaux
- la fonction *policy*, au coeur d'OpenAI

La *policy*, coeur de la stratégie

$$\begin{array}{rccc} \pi_\theta : & O & \rightarrow & \mathcal{D}(A) \\ & o & \mapsto & \mathcal{L} \end{array}$$

où :

- O est l'historique des observations
- A est l'espace des actions possibles
- $\mathcal{D}(A)$ est l'ensemble des distributions de probabilité sur A
- $\mathcal{L}$ est une distribution de probabilité sur A

et θ est le vecteur des paramètres (150 millions) qui définissent π, déterminés par un **réseau de neurones**.

Fonctionnement général

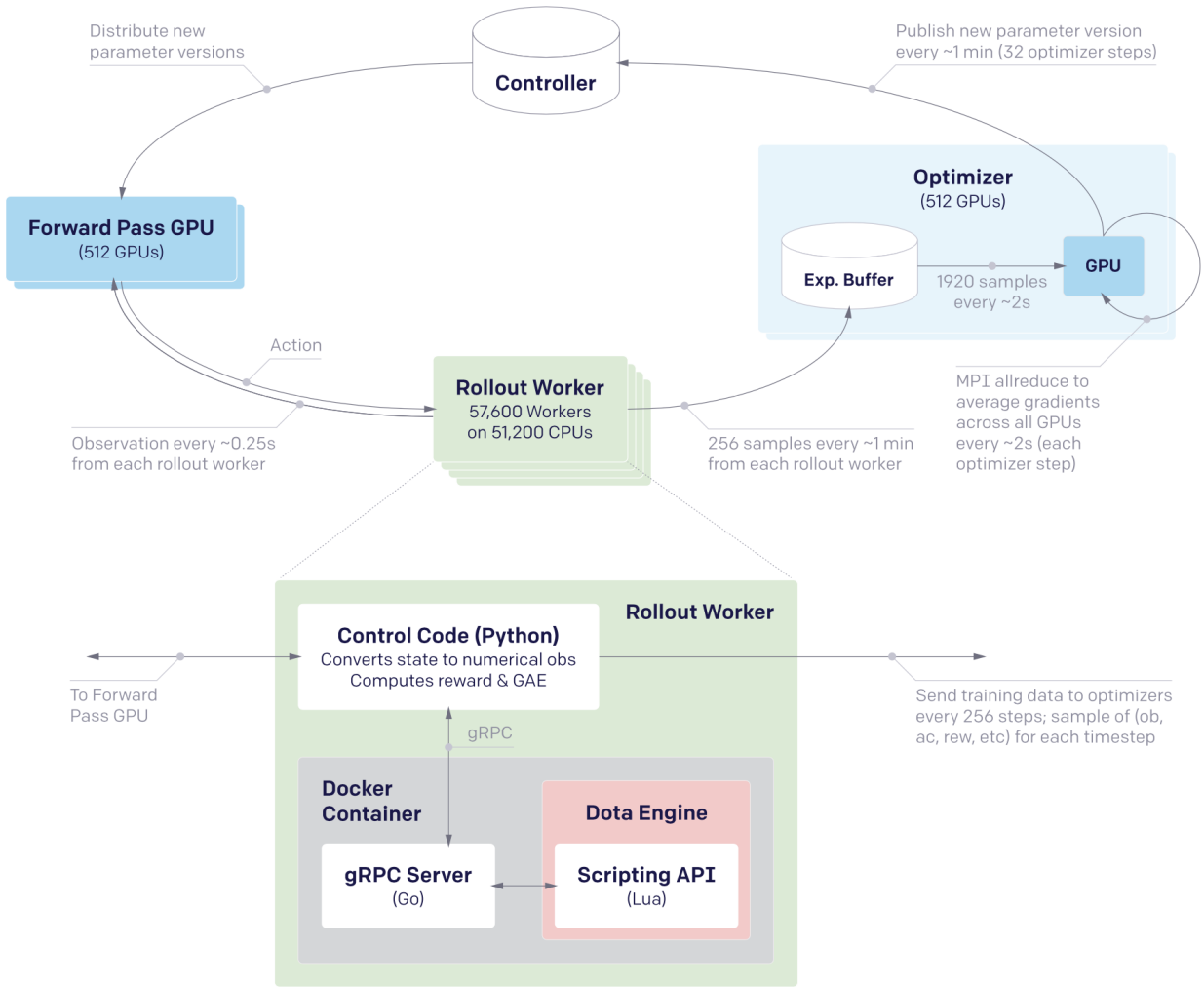


Figure: Fonctionnement général d'OpenAI

Un espace d'observation partiel

Seule la partie de la carte proche de l'unité du joueur et de ses bâtiments lui est visible. Or la prise de décision doit prendre en compte le comportement ennemi.

- Nécessité d'inférer le comportement ennemi.

Solution : $\pi_\theta : O \rightarrow \mathcal{D}(A)$

La nécessité d'une vision à long terme

20000 actions/partie, qui peuvent chacune avoir des conséquences considérables sur le reste de la partie.

Solution : Valuation des récompenses qui permet au réseau de neurones d'apprendre à investir sur des horizons temporels plus lointains.

Table of Contents

Évaluation des compétences d'un agent - Problème

Problème

Le système doit être capable d'évaluer automatiquement les compétences des agents au cours de leur formation.

Le système d'évaluation devra :

- Fonctionner pour des 1vs1 jusqu'à des 5vs5.

Évaluation des compétences d'un agent - Problème

Problème

Le système doit être capable d'évaluer automatiquement les compétences des agents au cours de leur formation.

Le système d'évaluation devra :

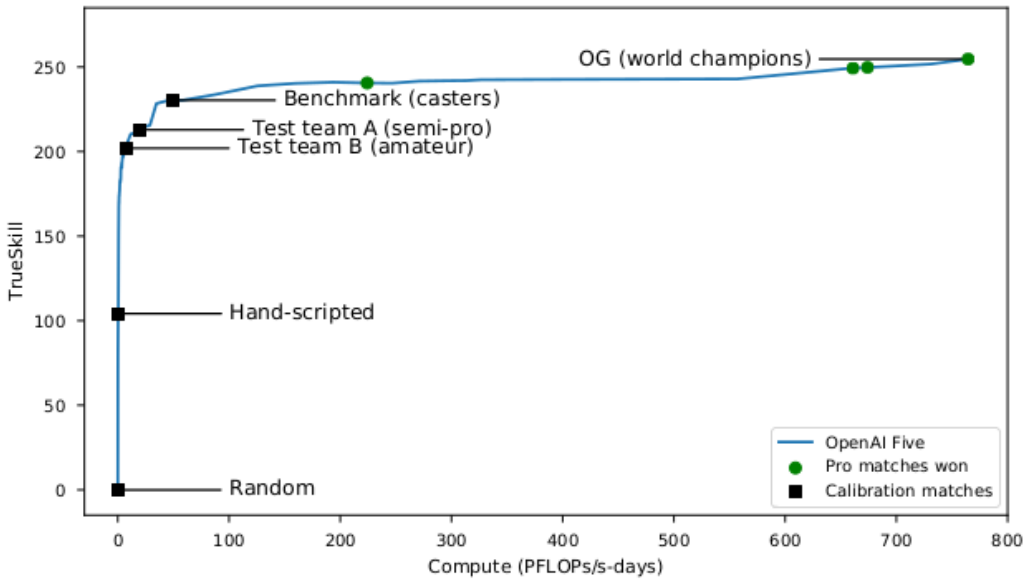
- Fonctionner pour des 1vs1 jusqu'à des 5vs5.
- Converger très rapidement vers les réelles compétences des agents.

Évaluation des compétences d'un agent - Solution

Solution

L'équipe de OpenAI a comparé les agents à un ensemble d'agents de référence fixes, dont les compétences sont connues et proches des agents testés, grâce au système d'évaluation TrueSkill.

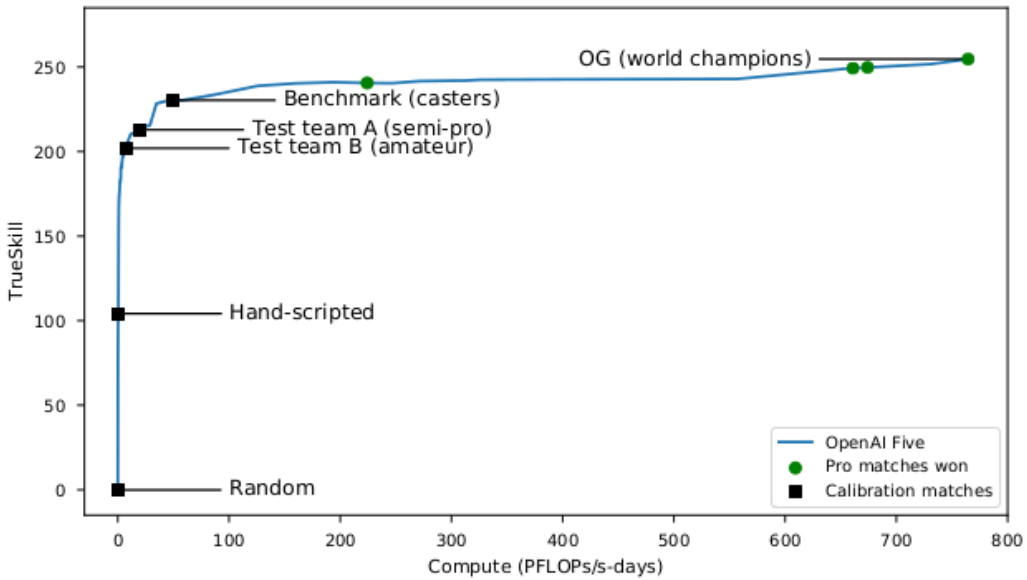
Évaluation des compétences d'un agent - Solution



- TrueSkill = 0 (agent aléatoire)

Figure: Évolution du TrueSkill durant la formation des agents d'OpenAI Five

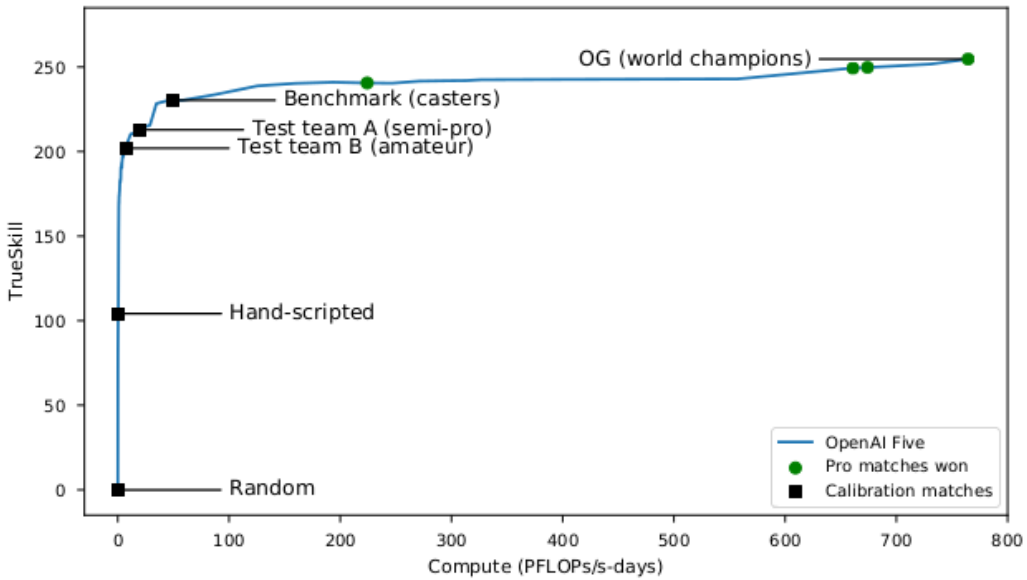
Évaluation des compétences d'un agent - Solution



- TrueSkill = 0 (agent aléatoire)
- TrueSkill = 105 (l'agent bat des débutants)

Figure: Évolution du TrueSkill durant la formation des agents d'OpenAI Five

Évaluation des compétences d'un agent - Solution



- TrueSkill = 0 (agent aléatoire)
- TrueSkill = 105 (l'agent bat des débutants)
- TrueSkill = 254 (l'agent bat des champions du monde)

Figure: Évolution du TrueSkill durant la formation des agents d'OpenAI Five

Évaluation des compétences d'un agent - Solution

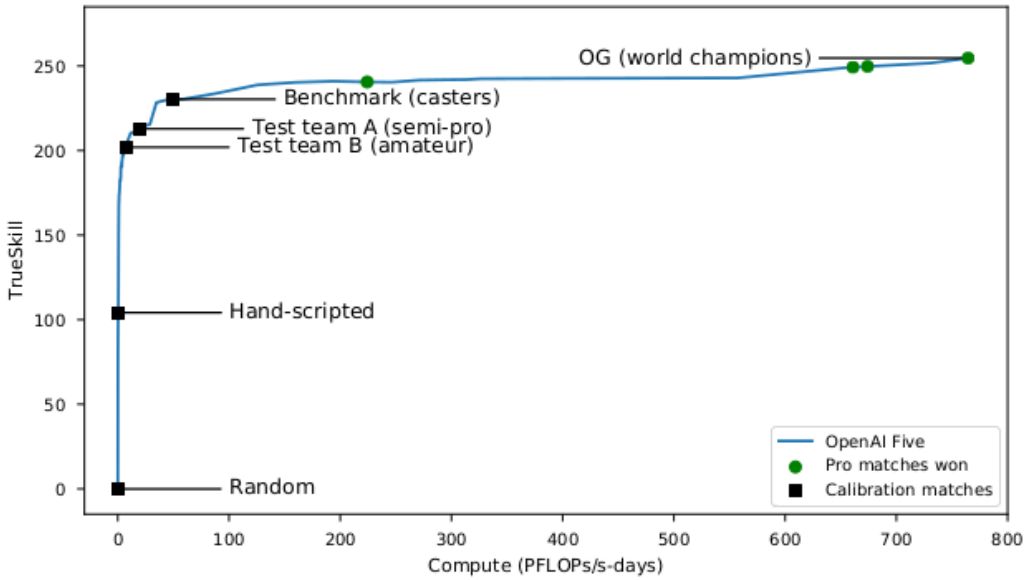


Figure: Évolution du TrueSkill durant la formation des agents d'OpenAI Five

- TrueSkill = 0 (agent aléatoire)
- TrueSkill = 105 (l'agent bat des débutants)
- TrueSkill = 254 (l'agent bat des champions du monde)
- Différence de TrueSkill = 10 (85% de chance de victoire pour l'agent avec le plus de TrueSkill)

Table of Contents

Temps de formation des agents - Problème

Problème

Le temps de formation de 10 mois des agents pose de nombreux problèmes concernant l'évolutivité du système puisqu'au cours des 10 mois le système subira inévitablement des changements sur les points ci-dessous.

- Processus de formation (structure de récompense, observations)

Temps de formation des agents - Problème

Problème

Le temps de formation de 10 mois des agents pose de nombreux problèmes concernant l'évolutivité du système puisqu'au cours des 10 mois le système subira inévitablement des changements sur les points ci-dessous.

- Processus de formation (structure de récompense, observations)
- Architecture de la "policy neural network"

Temps de formation des agents - Problème

Problème

Le temps de formation de 10 mois des agents pose de nombreux problèmes concernant l'évolutivité du système puisqu'au cours des 10 mois le système subira inévitablement des changements sur les points ci-dessous.

- Processus de formation (structure de récompense, observations)
- Architecture de la "policy neural network"
- Espace d'observations et espace d'actions des agents

Temps de formation des agents - Problème

Problème

Le temps de formation de 10 mois des agents pose de nombreux problèmes concernant l'évolutivité du système puisqu'au cours des 10 mois le système subira inévitablement des changements sur les points ci-dessous.

- Processus de formation (structure de récompense, observations)
- Architecture de la "policy neural network"
- Espace d'observations et espace d'actions des agents
- Mise à jour du jeu Dota 2 par l'éditeur Valve (héros, items, carte...)

Temps de formation des agents - Problème

Problème

Le temps de formation de 10 mois des agents pose de nombreux problèmes concernant l'évolutivité du système puisqu'au cours des 10 mois le système subira inévitablement des changements sur les points ci-dessous.

- Processus de formation (structure de récompense, observations)
- Architecture de la "policy neural network"
- Espace d'observations et espace d'actions des agents
- Mise à jour du jeu Dota 2 par l'éditeur Valve (héros, items, carte...)
- Tout changement entraîne l'interruption de la formation. Nous devons recommencer à zéro.

Temps de formation des agents - Solution

Solution

La technique qui va faciliter les modifications s'appelle "Surgery".

- N'interrompt pas la formation des agents

Temps de formation des agents - Solution

Solution

La technique qui va faciliter les modifications s'appelle "Surgery".

- N'interrompt pas la formation des agents
- Ajouter des paramètres (environnement, architecture du modèle, espace d'actions et d'observations) sans diminuer les compétences des agents (objectif)

Temps de formation des agents - Solution

Solution

La technique qui va faciliter les modifications s'appelle "Surgery".

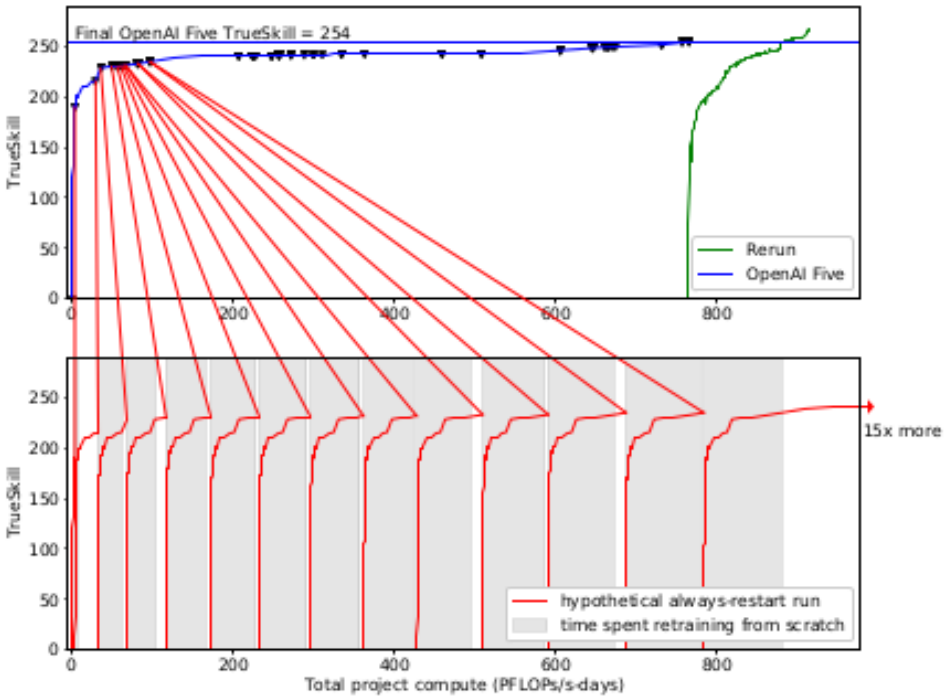
- N'interrompt pas la formation des agents
- Ajouter des paramètres (environnement, architecture du modèle, espace d'actions et d'observations) sans diminuer les compétences des agents (objectif)
- Surgeries apportées toutes les 2 semaines environ

Temps de formation des agents - Solution

Historique des plus importantes "surgeries"

Date	Iteration	# params	Change
6/30/2018	1	43,436,520	Experiment started
8/17/2018	81,821	43,559,322	Dota 2 version 7.19 adds new items, abilities, etc.
8/18/2018	84,432	43,805,274	Change environment to single courier; remove "cheating" observations
8/26/2018	91,471	156,737,674	Double LSTM size
9/27/2018	123,821	156,809,485	Support for more heroes
10/3/2018	130,921	156,809,501	Obs: Roshan spawn timing
10/12/2018	140,402	156,811,805	Item: Bottle
10/19/2018	144,121	156,286,925	Obs: Stock counts; Obs: Remove some obsolete obs
10/24/2018	150,111	156,286,867	Obs: Neutral creep & rune spawn timers
11/7/2018	161,482	156,221,309	Obs: Item swap cooldown; Obs: Remove some obsolete obs
11/28/2018	185,749	156,221,669	Item: Divine rapier; Obs: Improve observation of stale enemy heroes
12/10/2018	193,701	157,378,165	Obs: Modifiers on nonhero units.
12/14/2018	196,800	157,650,795	Action: Consumables on allies; Obs: Line of sight information; Obs: next item this hero will purchase; Action: buyback
12/20/2018	203,241	157,679,655	Dota 2 version 7.20 adds new items, new item slot, changes map, etc; Obs: number of empty inventory slots

Temps de formation des agents - Solution



- Graphe bleu : Évolution du TrueSkill avec la "surgery"
- Graphe rouge : Évolution du TrueSkill sans la "surgery" (recommencer à zéro)
- Graphe vert : Évolution du TrueSkill après un Rerun de la version finale d'OpenAI Five
- **TrueSkill** : système de classement qui permet d'évaluer automatiquement les agents (105 - débutant / 254 - champion du monde)

Optimisation de OpenAI Five - Le problème

Problème

Améliorer les performances de sorte à obtenir les meilleurs performances possibles en un temps d'entraînement le plus court possible.

- Quels paramètres faire varier pour accélérer l'apprentissage ?

Optimisation de OpenAI Five - Le problème

Problème

Améliorer les performances de sorte à obtenir les meilleurs performances possibles en un temps d'entraînement le plus court possible.

- Quels paramètres faire varier pour accélérer l'apprentissage ?
- Quels paramètres faire varier pour obtenir l'IA avec les meilleures performances possible ?

Optimisation de OpenAI Five - Le problème

Problème

Améliorer les performances de sorte à obtenir les meilleurs performances possibles en un temps d'entraînement le plus court possible.

- Quels paramètres faire varier pour accélérer l'apprentissage ?
- Quels paramètres faire varier pour obtenir l'IA avec les meilleures performances possible ?
- Trois axes d'optimisation : Batch Size, Data Quality et Long Term Credit Assignement

Optimisation de OpenAI Five - Batch Size

Problème : accélérer l'apprentissage

Quel impact a la taille de lot d'entraînement sur la qualité et rapidité d'entraînement de l'IA ?

Optimisation de OpenAI Five - Batch Size

Solution

Afin d'accélérer le processus d'apprentissage il faut maximiser le batch size.

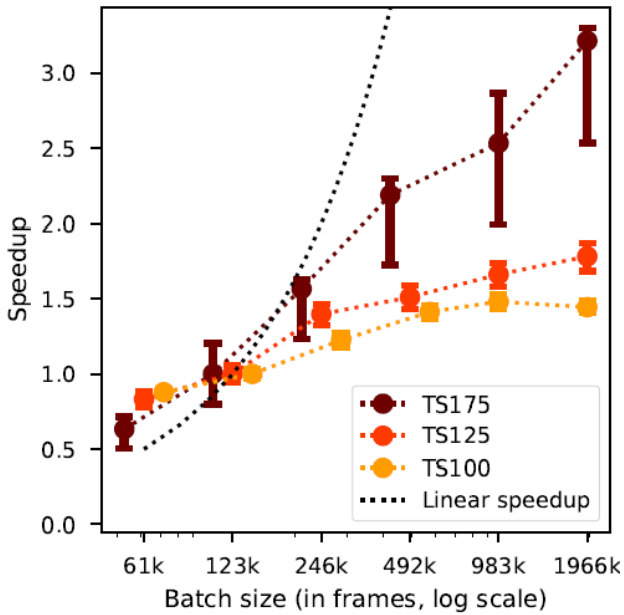


Figure: Accélération de l'entraînement en fonction de la taille du Batch Size

Optimisation de OpenAI Five - Batch Size

Autre conséquence

En maximisant la taille de lots, on améliore également les performances de l'IA produite

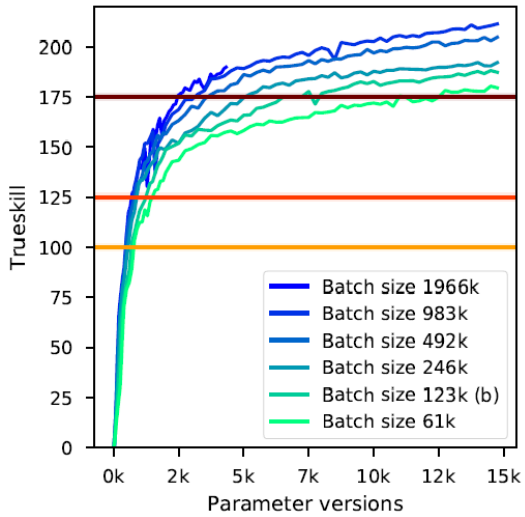


Figure: Amélioration des performances de l'IA en fonction du batch size

Optimisation de OpenAI Five - Data Quality

Problème : Améliorer l'apprentissage

La durée d'une partie et le fonctionnement asynchrone des composants entraînent de multiples mises à jour de la policy au cours d'une partie. Ce qui pose des questions de qualité des données.

- Existence de plusieurs versions de l'IA en parallèle.

Optimisation de OpenAI Five - Data Quality

Problème : Améliorer l'apprentissage

La durée d'une partie et le fonctionnement asynchrone des composants entraînent de multiples mises à jour de la policy au cours d'une partie. Ce qui pose des questions de qualité des données.

- Existence de plusieurs versions de l'IA en parallèle.
- Les mêmes échantillons de données peuvent être utilisés plusieurs fois au cours de l'apprentissage.

Optimisation de OpenAI Five - Data Quality

Solution : Staleness

Staleness défini comme

$$M - N$$

M la version de l'échantillon en cours d'optimisation
N la version à laquelle l'échantillon a été produite

- Afin de contrôler l'existence de plusieurs versions en parallèle.

Optimisation de OpenAI Five - Data Quality

Solution : Staleness

Staleness défini comme

$$M - N$$

M la version de l'échantillon en cours d'optimisation
N la version à laquelle l'échantillon a été produite

- Afin de contrôler l'existence de plusieurs versions en parallèle.
- Si ≥ 8, apprentissage significativement ralenti.

Optimisation de OpenAI Five - Data Quality

Solution : Staleness

Staleness défini comme

$$M - N$$

M la version de l'échantillon en cours d'optimisation
N la version à laquelle l'échantillon a été produite

- Afin de contrôler l'existence de plusieurs versions en parallèle.
- Si ≥ 8, apprentissage significativement ralenti.
- Vise staleness $\in [0, 1]$

Optimisation de OpenAI Five - Data Quality

Solution : Staleness

Staleness défini comme

$$M - N$$

M la version de l'échantillon en cours d'optimisation
N la version à laquelle l'échantillon a été produite

- Afin de contrôler l'existence de plusieurs versions en parallèle.
- Si ≥ 8, apprentissage significativement ralenti.
- Vise staleness $\in [0, 1]$
- Envoi données rollouts workers vers Optimizer toutes les 30s.
- MàJ des rollouts workers toutes les minutes.

Optimisation de OpenAI Five - Data Quality

Solution : Sample Reuse

Sample Reuse défini comme

$$\frac{\text{vitesse consommation données}}{\text{vitesse production données}}$$

- Afin de contrôler la réutilisation des échantillons.

Optimisation de OpenAI Five - Data Quality

Solution : Sample Reuse

Sample Reuse défini comme

$$\frac{\text{vitesse consommation données}}{\text{vitesse production données}}$$

- Afin de contrôler la réutilisation des échantillons.
- Si ratio $\in [2, 3]$, apprentissage 2 fois plus lent.

Optimisation de OpenAI Five - Data Quality

Solution : Sample Reuse

Sample Reuse défini comme

$$\frac{\text{vitesse consommation données}}{\text{vitesse production données}}$$

- Afin de contrôler la réutilisation des échantillons.
- Si ratio $\in [2, 3]$, apprentissage 2 fois plus lent.
- Ratio $\simeq 8$ empêche entraînement policy compétente.

Optimisation de OpenAI Five - Data Quality

Solution : Sample Reuse

Sample Reuse défini comme

$$\frac{\text{vitesse consommation données}}{\text{vitesse production données}}$$

- Afin de contrôler la réutilisation des échantillons.
- Si ratio $\in [2, 3]$, apprentissage 2 fois plus lent.
- Ratio $\simeq 8$ empêche entraînement policy compétente.
- Système vise ratio $= 1$.

Optimisation de OpenAI Five - Long term credit assignement

Problème : améliorer les performances

Besoin d'établir un plan en jeu pour maximiser le taux de victoire sur des parties d'environ 20k timesteps.

- Rentabiliser les ressources de sorte à obtenir les récompenses les plus intéressantes.

Solution : Long term credit assignement

Assigner les ressources pour maximiser le gain sur l'horizon de temps le plus important possible.

Optimisation de OpenAI Five - Long term credit assignement

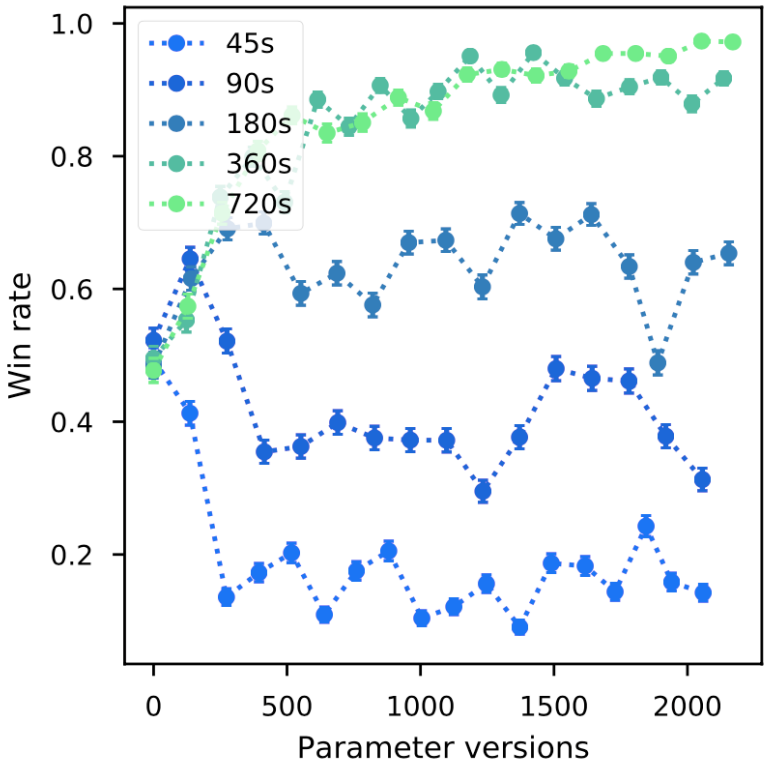


Figure: Évolution du winrate de l'IA en fonction de la version des paramètres et de l'horizon exploré

- Augmenter l'horizon augmente le winrate.

Optimisation de OpenAI Five - Long term credit assignement

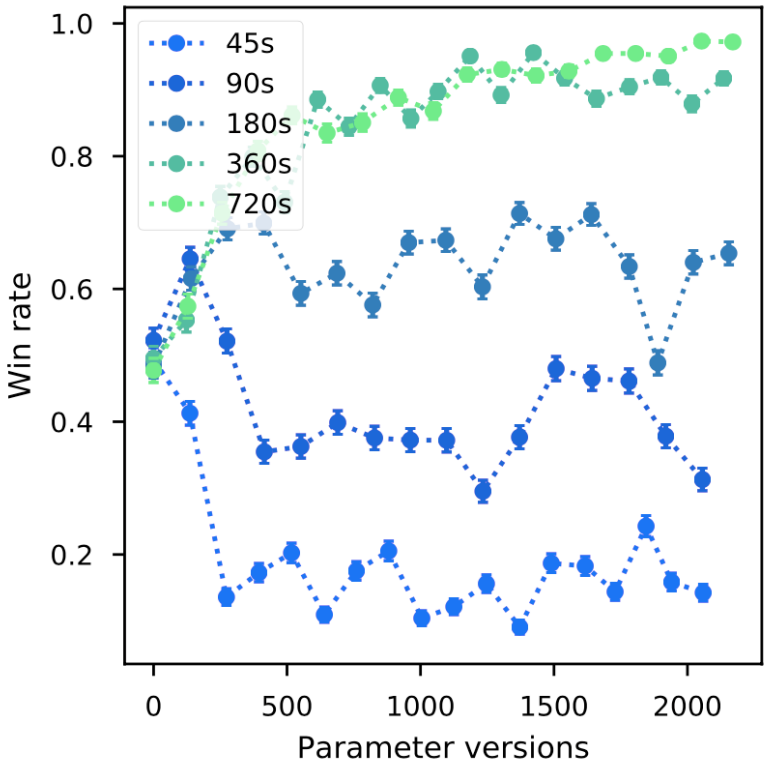


Figure: Évolution du winrate de l'IA en fonction de la version des paramètres et de l'horizon exploré

- Augmenter l'horizon augmente le winrate.
- Horizon de 180s pour l'agent qui a battu les champions du monde.

Optimisation de OpenAI Five - Long term credit assignement

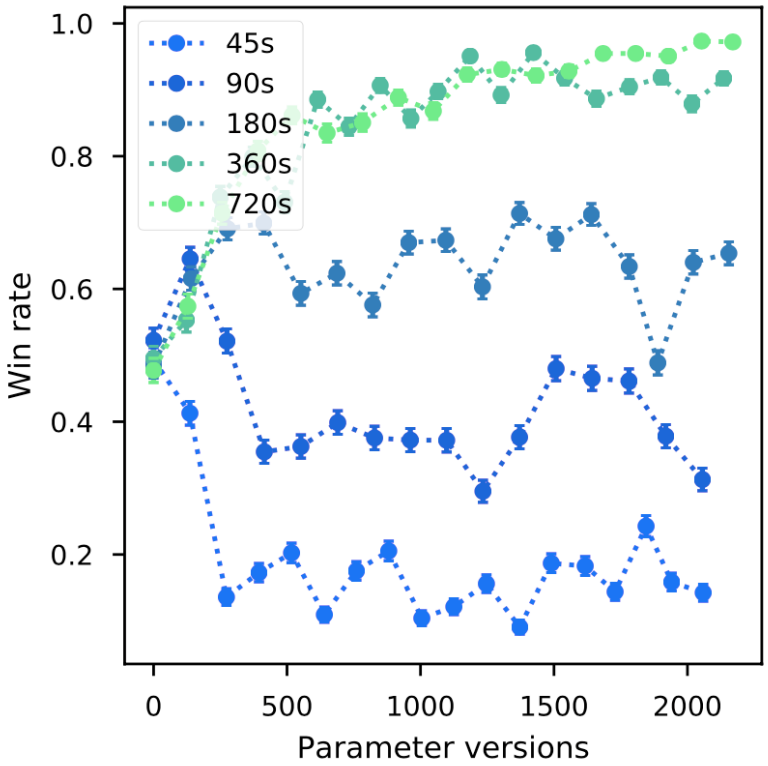


Figure: Évolution du winrate de l'IA en fonction de la version des paramètres et de l'horizon exploré

- Augmenter l'horizon augmente le winrate.
- Horizon de 180s pour l'agent qui a battu les champions du monde.
- Horizon exploré jusqu'aux 12 prochaines minutes.

Table of Contents

Limites d'OpenAI Five

- Temps de réponse

Limites d'OpenAI Five

- Temps de réponse
- Simplification du jeu

Limites d'OpenAI Five

- Temps de réponse
- Simplification du jeu
- Performance des surgeries

Limites d'OpenAI Five

- Temps de réponse
- Simplification du jeu
- Performance des surgeries
- Impact des hyperparamètres

Limites d'OpenAI Five

- Temps de réponse
- Simplification du jeu
- Performance des surgeries
- Impact des hyperparamètres
- Coût d'utilisation et d'optimisation croissant

Conclusion

Ce sera tout pour nous, Merci de votre attention !