Profielwerkstuk Voorstel:

Reinforcement Learning en Computerspellen

Hoe beïnvloeden de specifieke kenmerken van computerspellen de effectiviteit van verschillende reinforcement learning-algoritmes in het optimaliseren van spelprestaties?

Matthijs Gorter
Thom Brinkhorst
Pepijn van Iperen

Profielwerkstuk
onder begeleiding van
S. Rook
Christelijk Lyceum Zeist
Natuur en Techniek
6 september 2024

1 Doel van het onderzoek

Het doel van dit onderzoek is om te begrijpen hoe de kenmerken van verschillende computerspellen de effectiviteit van verschillende reinforcement learning (RL) algoritmes beïnvloeden bij het verbeteren van spelprestaties. Dit onderzoek richt zich op het identificeren van de eigenschappen van verschillende soorten spellen en de kenmerken van RL-algoritmes.

Door verschillende RL-algoritmes toe te passen op een reeks spellen met verschillende kenmerken, willen we ontdekken welke algoritmes het beste presteren in welke soorten spellen. Dit kan variëren van strategische spellen die planning vereisen tot actiespellen die snelle beslissingen vragen.

2 Onderzoeksvragen

2.1 Hoofdvraag

Hoe beïnvloeden de specifieke kenmerken van computerspellen de effectiviteit van verschillende reinforcement learning-algoritmes in het optimaliseren van spelprestaties?

2.2 Deelvragen

Om beter te begrijpen hoe de kenmerken van computerspellen de prestaties van verschillende reinforcement learning (RL) algoritmes beïnvloeden, hebben we drie belangrijke deelvragen opgesteld

1. **Wat zijn de specifieke kenmerken van verschillende soorten computerspellen?**

   Deze vraag richt zich de eigenschappen van verschillende soorten computerspellen. Spellen kunnen sterk verschillen in hoe ze zijn opgebouwd, hoe snel spelers beslissingen moeten nemen en hoe complex de spelregels zijn. Door deze kenmerken te onderzoeken, kunnen we inzicht krijgen in welke aspecten van een spel een uitdaging vormen voor RL-algoritmes.

2. **Welke reinforcement learning-algoritmes zijn beschikbaar en wat zijn hun kenmerken?**

   Hier willen we kijken naar de verschillende soorten RL-algoritmes die beschikbaar zijn en wat hen uniek maakt. Sommige algoritmes zijn beter in het leren van eenvoudige taken, terwijl andere juist goed zijn in het omgaan met complexe situaties.

3. **Hoe beïnvloeden de spelkenmerken de prestatie van reinforcement learning-algoritmes?**

   Deze vraag gaat in op het belangrijkste deel van het onderzoek: het verband tussen de kenmerken van een spel en hoe goed een RL-algoritme presteert. We willen weten hoe bepaalde eigenschappen van een spel, zoals de noodzaak voor snelle beslissingen of lange-termijnplanning, invloed hebben op de effectiviteit van een algoritme. Door de prestaties van verschillende algoritmes in verschillende spellen te vergelijken, kunnen we ontdekken welke het beste werken voor bepaalde soorten spellen en waarom dat zo is.

3 Experimenten

In dit onderzoek willen we kijken hoe verschillende reinforcement learning (RL) algoritmes presteren in verschillende computerspellen. We hebben drie spellen gekozen: Snake, Schaken, en Mario Super Bros. Elk spel heeft zijn eigen kenmerken en uitdagingen, en we zullen drie RL-algoritmes testen: Deep Q-Network (DQN), Proximal Policy Optimization (PPO), en AlphaZero (of Deep Deterministic Policy Gradient (DDPG) als AlphaZero niet lukt). Het doel is om te ontdekken hoe goed elk algoritme werkt in elk spel en hoe de kenmerken van het spel de prestaties van de algoritmes beïnvloeden.

1. **Snake**

   Snake is een simpel actiespel waar je een slang bestuurt die appels eet om langer te worden, terwijl je ervoor zorgt dat hij zichzelf niet raakt. Het spel vereist snelle beslissingen en het vooruitzicht zodat je je zelf niet opsluit. Het doel van het experiment met Snake is om te onderzoeken hoe de RL-algoritmes presteren in een omgeving met beperkte ruimte en snel veranderende situaties. We zullen kijken hoe snel elk algoritme leert, hoe stabiel de prestaties zijn en wat de uiteindelijke score is.

2. **Schaken**

   Schaken is een complex bordspel met een grote hoeveelheid mogelijke zetten en uitkomsten. Dit experiment is bedoeld om te bepalen hoe de RL-algoritmes omgaan met de grote hoeveelheid mogelijkheden en de noodzaak om lange-termijnstrategieën te ontwikkelen. We zullen meten hoe snel de algoritmes leren, hoe goed de strategieën zijn die ze ontwikkelen (gecontroleerd door een schaakprogramma), en hoe consistent de prestaties zijn in verschillende schaaksituaties.

3. **Mario Super Bros**

   Mario Super Bros, is een platformspel waarin de speler een personage bestuurt dat door verschillende levels moet navigeren, obstakels moet vermijden en vijanden moet verslaan. Dit spel combineert elementen van actie en planning, met veel variatie in speelomgevingen en uitdagingen. Het doel van het experiment met Mario Super Bros is om te onderzoeken hoe de RL-algoritmes presteren in een dynamische omgeving waar zowel snelheid als strategie nodig zijn. We zullen beoordelen hoe snel de algoritmes leren, hoe stabiel de prestaties zijn, en hoeveel levels succesvol worden voltooid.

4 Hypothese

We verwachten dat:

1. Deep Q-Network het beste zal presteren in Snake omdat het algoritme snel kan leren in omgevingen met beperkte ruimte en snel veranderende situaties, waar directe beloningen een grote rol spelen.
2. Proximal Policy Optimization zal beter presteren in Mario Super Bros, omdat dit algoritme geschikt is voor dynamische omgevingen en situaties waar zowel snelheid en planning belangrijk zijn.
3. AlphaZero zal beter zijn in Schaken, vanwege het planning en lange-termijnstrategie die nodig zijn.

5 Relevantie van het Onderzoek

Dit onderzoek laat effectiviteit van reinforcement learning algoritmes in verschillende omgevingen laat zien, wat bijdraagt aan het beter gebruik van AI-systemen. Deze

kennis kan niet alleen worden toegepast binnen de game-industrie, maar ook in andere sectoren zoals de gezondheidszorg, zelfrijdende auto's en robotica.

6 Achtergrondinformatie

Reinforcement Learning (RL) is een tak binnen kunstmatige intelligentie waarin een agent leert door interactie met zijn omgeving. Een agent is een entiteit die leert en acties onderneemt. Bij een zelfrijdende auto is het besturingssysteem de agent, en bij een schaakspel is de schaker de agent. De omgeving is alles waarmee de agent interageert en die reageert op de acties van de agent. Bij een zelfrijdende auto is dit de weg waar de auto op rijdt en de voertuigen om de auto heen. Bij een schaakspel is dit het schaakbord.

De agent leert door interactie met zijn omgeving. De agent ontvangt beloningen of straffen (negatieve beloningen) als gevolg van zijn acties. Het doel van de agent is om een strategie te ontwikkelen die de cumulatieve beloning maximaliseert over tijd. Bij Super Mario Bros (1985) is Mario de agent en de agent krijgt een beloning als Mario richting de eindslag beweegt (meestal naar rechts) en als Mario een coin of een power up oppakt en Mario krijgt een straf als Mario sterft en hij krijgt elke seconde straf zodat hij zo snel mogelijk het level wilt voltooien.

6.1 Kerncomponenten van Reinforcement Learning

Een actie naar de beslissing die een agent neemt bij elke stap in een besluitvormingsproces. Acties worden aangeduid met a en worden gekozen uit een reeks mogelijke acties $\mathcal{A}$. Elke door de agent genomen actie beïnvloedt de interactie met de omgeving, wat leidt tot een verandering in de toestand en een daaruit voortvloeiende beloning.

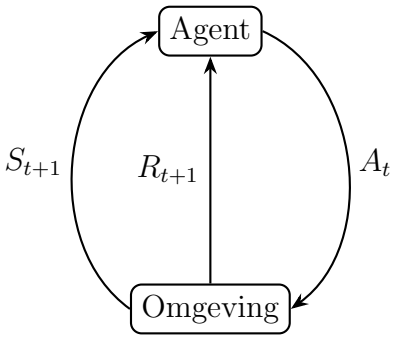
Een toestand x vertegenwoordigt de huidige situatie of staat van de omgeving waarin de agent opereert. Dit wordt aangeduid met x en maakt deel uit van de toestandsruimte $\mathcal{X}$. Bij de aanvangsstap $t = 0$, begint de agent in een initiële toestand x_0 die willekeurig wordt bepaald door een verdeling p. Naarmate het proces vordert, bevindt de agent zich in nieuwe toestanden gebaseerd op zijn acties.

Een beloning r is een feedbackwaarde die wordt ontvangen nadat de agent een actie heeft uitgevoerd in een bepaalde toestand. Deze beloning wordt bepaald door de beloningsfunctie $r(x, a)$. De beloningsmatrix R bevat de onmiddellijke beloningen voor elke combinatie van toestand en actie.

Een overgang beschrijft de verandering van de huidige toestand naar de volgende toestand als gevolg van een actie die door de agent wordt genomen. De

waarschijnlijkheid van overgang wordt bepaald door de overgangswaarschijnlijkheidsfunctie $p(x'|x, a)$, die afhangt van de huidige toestand x, de genomen actie a en leidt tot een nieuwe toestand x'. De overgangswaarschijnlijkheidsmatrix P bevat de waarschijnlijkheden van het overgaan van de ene toestand naar de volgende toestand, gegeven een bepaalde actie.

Figuur 1 illustreert het basismodel van interactie binnen Reinforcement Learning.



Figuur 1: Reinforcement learning interactiemodel tussen agent en omgeving via acties, toestanden en beloningen.

6.2 Markov Decision Process (MDP)

MDP werkt onder de Markov-aanname, wat betekent dat de volgende toestand en beloning alleen afhangen van het huidige toestand-actiepaar en niet van enige eerdere geschiedenis. Deze eigenschap vereenvoudigt het besluitvormingsmodel door zich alleen te concentreren op de huidige situatie.

Voorbeeld van een MDP:

- **Snake**: De toekomstige toestand (positie van de slang en voedsel) is volledig bepaald door de huidige toestand (huidige positie en locatie van het voedsel) en de actie (richting van beweging) zonder afhankelijk te zijn van de geschiedenis van eerdere bewegingen.

Voorbeeld van geen MDP:

- **Poker**: De beslissingen in poker zijn afhankelijk van niet alleen de huidige hand, maar ook van de geschiedenis van inzetten en het gedrag van andere spelers in vorige rondes.

In een MDP gaat een agent verder in tijdstappen $t = 0, 1, 2, \ldots, T$ waar de horizon T zowel eindig als oneindig kan zijn.

7 Onderzoeksplan en -overzicht

T = Gechatte tijd pp

Taak	**Persoon**	**T(uur)**	**Startdatum**	**Deadline**
Taakverdeling & planning	Alle	3	30/08/2024	04/09/2024
PWS voorstel maken	Alle	3	04/09/2024	06/09/2024
PWS voorstel inleveren	Thom	nvt	06/09/2024	06/09/2024
Inlezen in onderwerp	Alle	12	06/09/2024	20/09/2024
Theoretisch kader opstellen	Matthijs	20	20/09/2024	04/10/2024
Deelvraag 1 beantwoorden	Thom	10	20/09/2024	04/10/2024
Deelvraag 2 beantwoorden	Pepijn	10	20/09/2024	04/10/2024
Deelvraag 3 beantwoorden	Thom	15	04/10/2024	18/10/2024
Experiment 1 uitvoeren	Alle	15	04/10/2024	18/10/2024
Resultaten Experiment 1 verwerken	Pepijn	5	18/10/2024	25/10/2024
Experiment 2 uitvoeren	Alle	20	18/10/2024	25/10/2024
Herftvakantie en Toetsweek 1	Alle	nvt	25/10/2024	19/11/2024
Eerste versie PWS maken	Matthijs	10	20/11/2024	29/11/2024
Eerste versie inleveren	Thom	nvt	29/11/2024	29/11/2024
Resultaten Experiment 2 verwerken	Pepijn	5	20/11/2024	29/11/2024
Experiment 3 uitvoeren	Alle	15	29/11/2024	13/12/2024
Resultaten Experiment 3 verwerken	Pepijn	5	13/12/2024	20/12/2024
Kerstvakantie	Alle	nvt	21/12/2024	05/01/2025
Resultaten	Pepijn	5	05/01/2024	13/01/2025
Toetsweek 2	Alle	nvt	13/01/2024	31/01/2025
Conclusie	Thom	8	31/01/2024	16/02/2025
Onderzoeksmethode	Matthijs	3	31/01/2024	16/02/2025
Discussie	Thom	3	31/01/2024	16/02/2025
Voorwoord	Thom	3	31/01/2024	16/02/2025
Referentielijst	Pepijn	3	31/01/2024	16/02/2025
Nakijken van alle hoofdstukken	Alle	3	16/02/2024	19/02/2025
Alles samenvoegen in de eindversie	Matthijs	3	19/02/2024	21/02/2025
Eindeversie inleveren	Thom	nvt	21/02/2024	21/02/2025
Presentatie voorbereiden	Alle	5	21/02/2025	28/02/2025
Presentatie maken	Alle	5	28/02/2025	08/03/2025
Totaal	**Matthijs**	**97**	nvt	nvt
Totaal	**Thom**	**94**	nvt	nvt
Totaal	**Pepijn**	**93**	nvt	nvt
Totaal	**Alle**	**284**	nvt	nvt

8 Bronvermelding

Voor de theoretische onderbouwing en achtergrondinformatie zullen de volgende bronnen worden gebruikt:

- *Reinforcement Learning: An Introduction* door Andrew Barto en Richard S. Sutton.
  - https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf
  - http://incompleteideas.net/book/RLbook2020.pdf
- Stanford CS234 Winter 2019: *Reinforcement Learning* (15 colleges).
  - https://www.youtube.com/playlist?list=PLoROMvodv4rOSOPzutgyCTapiGlY2Nd8u
- *Spinning Up in Deep RL* door OpenAI.
  - https://spinningup.openai.com/en/latest/index.html
- Yunhao Tang 2021: *Deep Reinforcement Learning.*
  - https://web.archive.org/web/20240429102051/https://academiccommons.columbia.edu/doi/10.7916/d8-y0tc-h725/download
- *Proximal Policy Optimization Algorithms* door Schulman, J., Wolski, F., Dhariwal, P., Radford, A., en Klimov, O. (2017, 20 juli).
  - https://arxiv.org/abs/1707.06347
- *Playing Atari with Deep Reinforcement Learning* door Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., en DeepMind Technologies. (z.d.).
  - https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf
- *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm* door Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., en Hassabis, D. (2017, 5 december).
  - https://arxiv.org/abs/1712.01815

Daarnaast zullen eerdere PWS-projecten worden geraadpleegd, zoals:

- *Een kunstmatige opsporing van longkanker* (2023) - KNAW. https://storage.knaw.nl/2023-06/profielwerkstuk-2023-een-kunstmatige-opsporing-van-longkanker.pdf
- *File en nog eens file* (2023) - KNAW. https://storage.knaw.nl/2023-06/pws-file-file-en-nog-eens-file.pdf