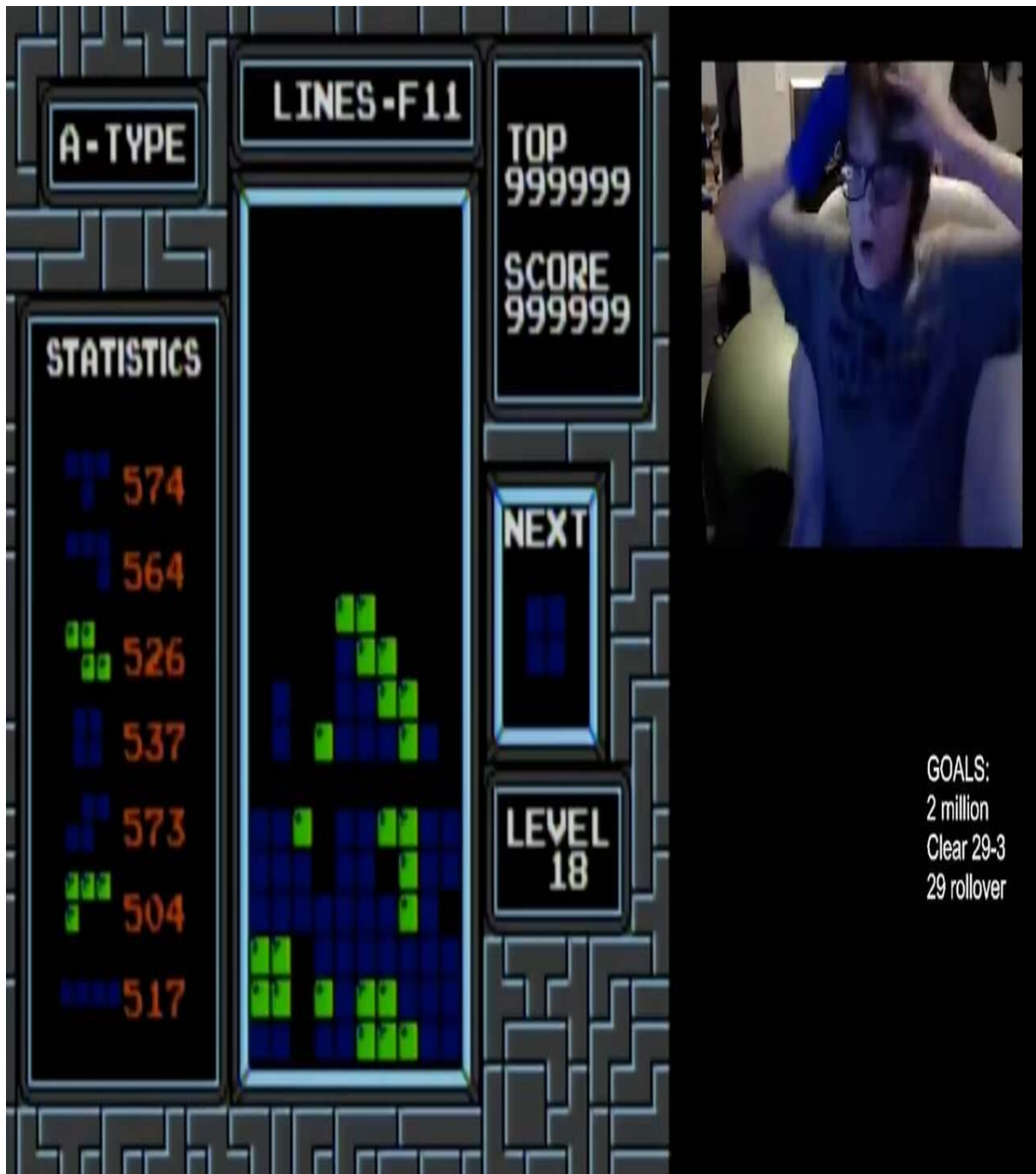


Reinforcement Tetris

28/03/2025



(Nws, 2024)

Gemaakt door Casper de Knecht, Mark Olieman en Sam Keesen

Probleemstelling

Het probleem

Tetris een simpel spel gemaakt in 1984 in de Sovjet-Unie door Alexey Pajitnov. Een spel dat snel 1 van de grootste spellen ooit werd. Velen attribueren het succes van tetris aan de oneindige mogelijkheden van het spel waardoor je elke keer een uniek spel speelt. Belangrijk is ook dat tetris een laag instapniveau heeft met makkelijke regels waardoor het aantrekkelijk is voor iedereen maar ook heeft het een hoog expertise ontwikkeling waardoor het door de diehard fans aantrekkelijk is. Tetris begint met een leeg veld. In dit lege veld vallen er constant blokken naar beneden die verschillende vormen kunnen zijn, denk bijvoorbeeld aan een vierkant of aan een T-vorm. Het speel veld kan je zien als een rooster met de dimensies van 10 kolommen bij 20 rijen. Als een rij gevuld is met blokjes worden alle blokjes in deze rij weggehaald. Hiervoor en voor alle blokjes die je plaats krijg je een bepaald hoeveelheid punten. (*Falling Into Place: Piecing Together Tetris' Enduring Legacy*, 2024)

Wij zijn groot tetris fans en waren benieuwd hoe goed wij een agent zouden kunnen maken om tetris te spelen.

Motivatatie voor een Reinforcement Learning systeem

Hoewel tetris een simpel spel is met simpele regels zijn er oneindige mogelijkheden voor nieuwe spelsituaties. Door de oneindig vele mogelijkheden is een rule-based systeem moeilijk te implementeren, hierbij zou je oneindig veel regels moeten bedenken voor de oneindig veel mogelijkheden. Dit is haast een onmogelijke taak waardoor het niet een toepasselijk systeem is voor dit probleem. Een mogelijk niet AI-oplossing is ook niet toepasbaar. Dit komt doordat je oneindig veel data zou moeten hebben om bijvoorbeeld een Machine Learning systeem toe te passen. Buiten de vele data zou dit ook veel rekenkracht en tijd kosten.

De principes van reinforcement learning past nauw bij de principes van het behalen van een hoge score in tetris. Er is bij tetris geen perfecte oplossing. Dit komt je niet weet welke blokken er daarna allemaal komen. Vaak is het ook dat de directe zet voor de hoogste aantal punten in de lange termijn voor minder punten zorgt. Hierdoor is een langetermijnstrategie erg van belang bij tetris. Dit zijn punten waar reinforcement learning erg goed in is. Reinforcement learning is ontworpen om lange termijn beloningen te maximaliseren en om goede policies te leren door ervaring. Hierom hebben wij gekozen voor een reinforcement learning systeem. (*ChatGPT - RL in Tetris*, z.d.)

Methodologie

Ontwerpkeuzes

We laden de tetris omgeving in via 'TetrisA-v3' van gym_tetris en vereenvoudigen die met 'JoypadSpace' waardoor we NES controls kunnen simuleren. We hebben een convolutioneel neural netwerk getraind om de beelden van het spel te verwerken. Hierbij is de output de Q-waarden voor elk mogelijke actie. We hebben een replay buffer toegepast om ervoor te zorgen dat je correlaties tussen opeenvolgende stappen vermijdt. Epsilon-Greedy strategie toegepast waarbij epsilon langzaam afbouwt van 1.0 naar 0.01. Dit is gebruikt omdat de agent aan het begin nog niks weet dus moet die uit gaan vinden wat de beste opties zijn. Wij hebben een reward functie gebouwd waarbij de agent punten krijgt wanneer die een lijn weghaalt en punten verliest als die gaten maakt, de blokken te hoog maakt of als hij verliest. Dit hebben we gedaan omdat de standard reward functie van alleen punten geven voor het weghalen van een lijn niet voor goede strategieën zorgt en ook nog eens heel schaars is. We hebben MSE gebruikt om de loss functie te berekenen omdat dat standaard is voor een regressieprobleem zoals hier de Q-waarde schatting. Na 50 episodes reset het environment, dit is om de stabiliteit van het script te verhogen over langere periodes. (*ChatGPT - DQN Voor Tetris Uitleg*, z.d.)

Implementatiedetails

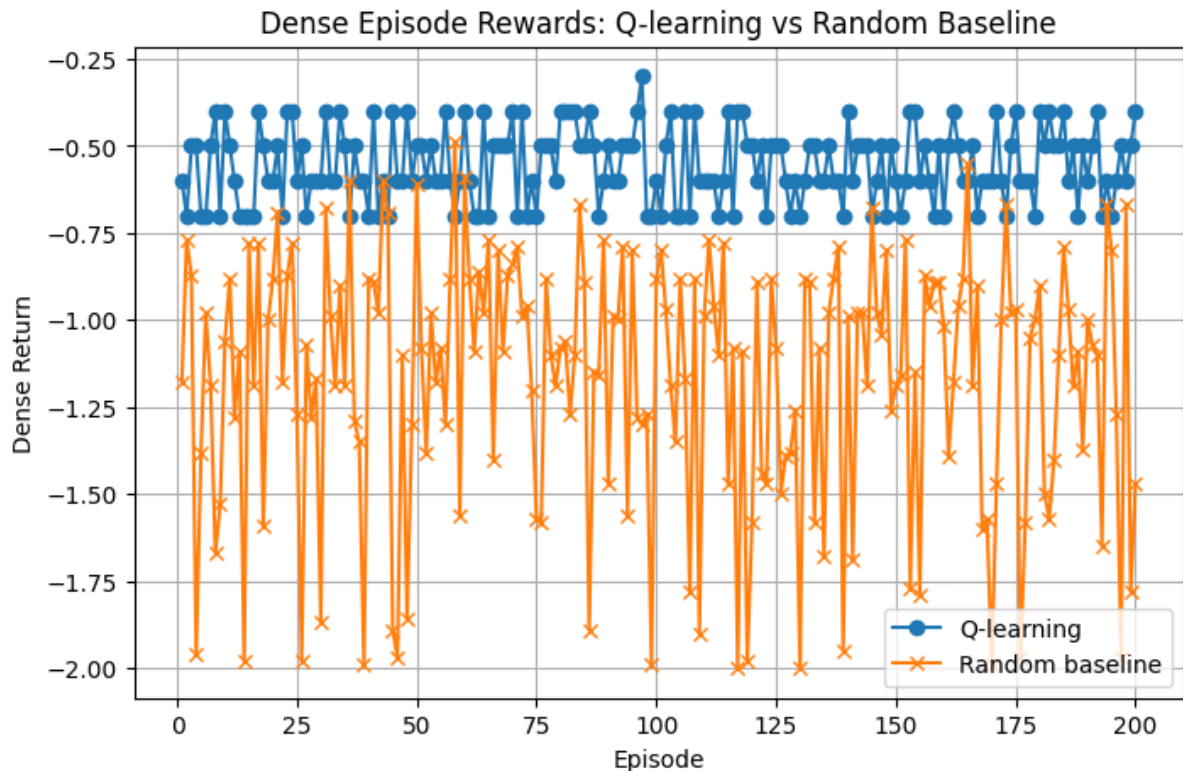
Ons reinforcement learning systeem is geïmplementeerd in Python en volgt een modulaire structuur:

- Main.ipynb: Een verdere uitleg van de code en het experiment.
- README.md: Uitleg over installatie en gebruik.
- requirements.txt: Lijst van benodigde Python-pakketten.

De implementatie is goed gedocumenteerd en reproduceerbaar, zodat anderen de bot eenvoudig kunnen uitvoeren en verbeteren. Voor meer informatie verwijzen we naar het Main.ipynb bestand.

Conclusie

We hebben 200 rondes gesimuleerd voor beide de baseline en de Q-learning. Hieruit is dit resultaat gekomen.



Je ziet dat ons Q-learning agent het daadwerkelijk wel beter doet dan een random baseline. Je ziet dat de random baseline elke keer een willekeurige negatieve return geeft. Dit is vanzelfsprekend want willekeurig iets doen geeft geen goed resultaat. Je ziet ook dat er niet echt een verbetering is gekomen in de Q-learning model dit komt door de lage hoeveelheid episodes.

Reflectie

De grootste valkuil voor ons in dit project was timemanagement. We zijn veel te laat begonnen met het uitvoeren van de opdracht waardoor er fouten zijn waar we niet genoeg tijd voor hebben om op te lossen. Het werk is hierdoor ook slordig en van een niveau waar we niet trots op zijn. Dit willen wij allemaal voor het volgende project vermijden. Daarbuiten is er mogelijk ook wat fouts gegaan in onze reward functie, er komen alleen maar negatieven rewards uit wat niet de bedoeling hoort te zijn. We denken persoonlijk dat er iets misgaat bij het berekenen van het verlies van punten bij gaten maar hebben dit niet kunnen controleren.

Bronnenlijst

Nws, V. (2024, 3 januari). Bijna 40 jaar na lancering speelt tiener “Tetris” helemaal uit: “Ik voel mijn vingers niet meer” | VRT NWS: nieuws. *VRTNWS*.

<https://www.vrt.be/vrtnws/nl/2024/01/03/tetris-uitspelen/>

Falling into place: piecing together Tetris' enduring legacy. (2024, 7 februari). Temple Now | news.temple.edu. [https://news.temple.edu/news/2023-03-22/falling-place-piecing-](https://news.temple.edu/news/2023-03-22/falling-place-piecing-together-tetris-enduring-legacy-0)

[together-tetris-enduring-legacy-0](https://news.temple.edu/news/2023-03-22/falling-place-piecing-together-tetris-enduring-legacy-0)

ChatGPT - RL in Tetris. (z.d.). ChatGPT. <https://chatgpt.com/share/681e6667-0edc-800f-ac49-87dc561db4de>

ChatGPT - DQN voor Tetris uitleg. (z.d.). ChatGPT. <https://chatgpt.com/share/681e663b-1754-800f-a08d-ac5743b4bd3a>