**Hearthstone AI: Prediksjon av beste valg ved bruk av MCTS og Supervised Machine Learning**

TDAT3025 Anvendt maskinlæring

Skrevet av Emir Derouiche

23.November 2020

# Introduksjon: Hvorfor Hearthstone?

## Maskinlæring og brettspill:

Maskinlæring blir er kjent for å bli anvendt for finne gode strategier for brettspill som sjakk og go hvor tradisjonelle framgangsmåter krever en ikke-triviell mengde prosesseringskraft. Slike spill har lenge vært en av de få resterende intelligenstestene hvor mennesker kan beseire datamaskiner. Dette kommer av menneskers evne til å filtrere ut ugunstige trekk selv når antall mulige kombinasjoner av trekk er ufattelig høye. Mennesker kan gjøre dette på grunn av vår evne til å lære fra erfaring og det er nettopp denne evnen vi prøver å gi til datamaskiner med maskinlæring.

Hearthstone, et moderne brettspill:

Hearthstone: Heroes of Warcraft er et moderne brettspill utviklet av dataspillprodusenten Blizzard Entertaiment. Hearthstone kan beskrives som en kombinasjon av sjakk og poker. Hearthstone ligner på sjakk fordi man har brikker på slagfeltet som kan angripe fiendens brikker, hvor strategisk valg av angrepsmål er relevant. Hearthstone har allikevel mye tilfeldighet og skjult informasjon slik som i poker ettersom spillerne trekker tilfeldige kort og ikke vet hva slags kort motstanderen har på hånden. Evnen til å forutse motstanderens kort og fremtidige trekk er det som skiller nybegynnere fra mestere[1]. Da Hearthstone kom ut i 2014[2] vekket det umiddelbart interesse hos maskinlæringsmiljøet. HearthSim[3] er en gruppe utviklere som slo seg sammen for å skape et eget miljø rundt Hearthstone hvor man kan dele verktøy og erfaringer som kan være nyttige for andre utviklere. For mitt prosjekt valgte jeg å bruke en av simulatorene HearthSim har utviklet: fireplace[4].

Inspirasjon og begrensninger:

Den opprinnelige tanken for dette prosjektet var å benytte Monte Carlo Search Tree (MCST) i kombinasjon med deep learning i likhet med Deepminds AlphaGo[5]. Jeg endte opp med å benytte supervised learning i stedet for deep learning. Det finnes flere steder hvor deep learning kan anvendes for å forbedre systemet mitt, men for dette prosjektet gikk jeg dessverre tom for tid før jeg kunne begynne å implementere slike optimaliseringer. Min bruk av MCST ble også annerledes fra Deepmind sin framgsmåte ettersom systemet mitt i stor grad ble simplere enn AlphaGo grunnet åpenbare tidsbegrensninger og teambegrensninger. Dette vil bli utdypet i diskusjonsdelen av denne artikkelen.

# Teori

Begrensnigner for tradisjonell Hearthstone AI

Hearthstone har en divers mengde problemer hvor maskinlæring kan anvendes, noe som er godt utdypet i artikkelen «The Many AI Challenges of Hearthstone»[6]. For dette prosjektet snevret jeg inn omfanget til å bare se på angrepsmål for brikkene til spilleren. Hearthstones innebygde AI har en tradisjonell framgangsmåte, men denne er i stor grad mangelfull, dette er årsaken til at maskinlæring kan anvendes for å gi bedre resultater. En viktig årsak til at en maskinlæringsalgoritme kan utkonkurrere Hearthstone AIen er at forgreningsfaktoren for Hearthstone er ufattelig høy. Dette kommer av at Hearthstone tillater flere handlinger per tur og mengden handlinger kan gå mot uendelig. Mengde kombinasjoner av handlinger per tur vokser i tillegg eksponentielt i forhold til antall handlinger. Hearhstones AI måtte dermed ha store begrensninger for å oppnå sanntidsevaluering av trekk. Et av de største begrensningene er at AIen kun ser på kombinasjoner av opp til to kort når det kommer til angrepsmål[1].

# Tidligere relevant arbeid

# Framgangsmåte

Yrdy

# Resultat

Yrdy

# Diskusjon og videre arbeid

Played cards could be more relevant than attack targets

# Konklusjon

Yrdy

# Referanser

1. <https://www.gdcvault.com/play/1020592/AI-Postmortem>
2. <https://en.wikipedia.org/wiki/Hearthstone>
3. <https://hearthsim.info/>
4. <https://github.com/jleclanche/fireplace>
5. <https://deepmind.com/research/case-studies/alphago-the-story-so-far>
6. <https://link.springer.com/article/10.1007/s13218-019-00615-z>