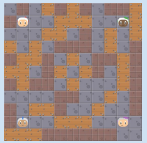
HVA ER EFFEKTENE AV REWARD SHAPING PÅ EN DQN AGENT I POMMERMAN?

Introduksjon

Pommerman, basert på det klassiske spillet Bomberman, er en multi-agent environment utfordring hvor man skal bruke maskinlæring til å slå de andre spillerne. I vårt tilfelle er disse agentene standard- agenten "SimpleAgent", en agent som har hardkodet oppførsel og handlinger. Det holdes også konkurranser om hvem som kan lage de beste agentene i Pommerman, både for FFA (Free-For-All) og for den lagbaserte varianten av Pommerman. Den første konkurransen ble holdt i 2018, hvor vinnerne Yichen Gong og Gorogm vant 4000 dollar. [1]



Pommerman-miljøet består av et brett/rutenett på størrelsen 11x11. Vegger og ødeleggbare bokser er plassert tilfeldig rundt på brettet og hver agent (opp til 4) starter runden i hvert sitt hjørnet av brettet. Agentene kan plassere ut bomber som etterlater seg flammer når de eksploderer. Bombene har en levetid på 10 steg, og vil ødelegge alt, utenom vegger, innenfor rekkevidden når de eksploderer. Disse kan brukes til å ødelegge bokser og til å drepe andre agenter. En runde stopper når det kun er én agent i livet. I tillegg er det også en maks antall steg som kan gjennomføres, og dersom dette antallet nås vil runden stoppe. [2]

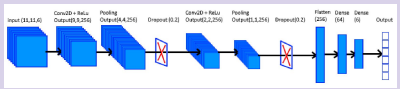
Vi har lagd en agent som bruker DQN, en type reinforcement learning, for å lære seg å manøvrere Pommerman-miljøet. For å undersøke problemstillingen: "Hva er effektene av reward shaping på en DQN agent i Pommerman?", tester vi agenten både med og uten reward shaping og sammenligner ytelsen og oppførselen.

Metode

Vi brukte Deep Q-Learning (DQN) for å trene agenten. DQN ble først utviklet for å trene agenter som foretar handlinger i en omgivelse med mål om å maksimere belønning. [3] Denne metoden fungerer altså bra når agenten har en begrenset diskret mengde handlinger den kan foreta. I Pommerman er vi ute etter å sende inn en "state", altså agentens observasjoner, med en bestemt størrelse og få ut verdier som forteller agenten hvilken handling (action) den burde ta. Agentene i Pommerman har også en ganske liten mengde mulige handlinger de kan foreta; stopp, opp, venstre, ned, høyre, og legg bombe. Dette er dermed noe DQN egner seg godt for.

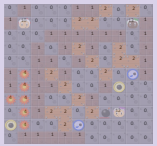
Det nevrale nettverket

Dette er oppsettet for det nevrale nettverket vi brukte for agentene:



Input

Størrelsen på input i det nevrale nettverket vi bruker i DQN er 11x11 og vi sender inn Pommermans state som vi former om til 6 lag. Hvert lag består av matriser eller enkeltverdier som representerer spillebrettet. Dataen vi har valgt å sende inn som input i det nevrale nettverket er "board", "bomb blast strength" , "bomb life", "bomb moving direction", "flame life", og "ammo". Med unntak av "ammo" så er alle disse verdiene lagret i 11x11 matriser. For å kunne sende verdien for ammunisjon inn i modellen sammen med de andre matrisene kreves det at de har samme dimensjon. Dette løste vi ved å plassere den i en nullmatrise med størrelsen 11x11.



Reward Shaping

Ved første trening av agenten, brukte vi standard reward-shape, altså +1 for seier og -1 for tap. Når vi skulle trene den med reward shaping, la vi til følgende belønningsverdier:

1. -0.005 poeng dersom agenten velger "stop"-handlingen eller at den ikke flytter på seg
2. +0.0005 dersom agenten er i en annen posisjon enn den var for et steg siden
3. +0.1 for hver fiende som er i nærheten av en plassert bombe
4. +0.05 for hver "wood" som er i nærheten av en plassert bombe
5. +0.001 for hver bombe plassert
6. +0.05 for hver rute agenten er unna en bombe

I tillegg til dette begrenset vi mengden belønning per steg for alle steg som ikke er terminale til å ligge i intervallet [-0.9, 0.9]. Dette ble gjort for å sørge for at en seier alltid er mest belønnende og et tap alltid mest straffende.

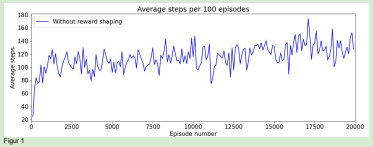
Resultater

Trening

Treningen av agenten ble gjennomført i 2 iterasjoner. Første gang uten reward shaping, så med reward shaping. Treningen av DQN-agenten ble gjennomført over 20000 episoder og den spilte mot 3 agenter av typen "SimpleAgent". Underveis i treningen lagret vi verdiene for antall steg tatt og belønning mottatt for hver episode.

Uten Reward Shaping

Først trente vi en agent uten reward shaping. Den eneste gangen agentene får en belønning er altså når en episode er over. Dette gjør treningen til agenten mer utfordrende, siden den ikke vil få en positiv eller negativ tilbakemelding for hver handlingen den foretar, noe som kan bidra til at agenten ikke utforsker nye muligheter. Figur 1 viser gjennomsnittlig antall steg tatt per 100 episoder. Helt i begynnelsen av treningen av agenten uten reward shaping kan man se en rask økning i antall steg tatt per episode grunnet en høyere utforskningsrate, men denne økningen avtar raskt etter hvert som valget av handlinger blir mindre tilfeldig. Men man kan allikevel se en liten økning i antall steg tatt etter hvert som treningen pågår. Gjennomsnittlig tar agenten omtrent 120 steg per episode, men det er en ganske stor variasjon i lengden på episodene.



Figur 1

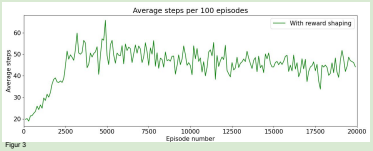
I motsetning til "SimpleAgent" som aktivt beveger seg rundt og "jakter" på andre agenter, så står den trente DQN-agenten for det meste helt stille. Dette er også observert av andre implementasjoner av Pommerman med DQN uten reward shaping, blant annet i en rapport om anvendelse av reinforcement learning til Pommerman. [4] Mangelen på tilbakemeldinger vil føre til at den lærer handlingen med å plassere bomber øker sjansen for å tape, da det er flere trusler i nærheten, i stedet for at handlingen øker sjansen for å vinne ved å jakte etter andre. Som nevnt tidligere, vil agenten kun få 1 eller -1 i belønning uten reward shaping, og med den apatiske strategien den utviklet vil den tape mesteparten av tiden. Dette fører derfor til at agentens gjennomsnittlige belønning vil ligge så vidt over -1 som vist på figur 2..

Figur 2

Med Reward Shaping

For å gi agenten flere tilbakemeldinger på dens handlinger underveis i treningen, la vi til reward shaping. Agenten ble igjen trent i 20000 episoder. Dette gir agenten direkte tilbakemelding på konkrete handlinger det utfører hvert steg av treningen, i stedet for å kun gi en tilbakemeldingen til agenten på slutten av en runde/episode. Ved trening av denne agenten så øker gjennomsnittlig antall steg tatt gjennom hele utforsknings-perioden, som man kan se på figur 4, siden agenten prøver ut nye handlinger og lærer hva som fører til høyest belønning.

Etter at denne perioden er over og handlingene til agenten stort sett blir bestemt av Q-verdiene fra modellen, stopper økningen av episode-lengden, og agenten "setter seg fast" i en bestemt oppførsel. Ettersom agenten nå lærer seg å benytte seg av flere av de tilgjengelige handlingene, vil den også utsette seg selv for høyere risiko. Dette førte til, som kan sees i figur 3, til at det gjennomsnittlige antallet steg tatt hver episode er ganske mye lavere enn det var for agenten uten reward shaping.



Figur 3

Med reward shaping får agenten et større insentiv til å utforske, da den blir straffet for å stå i ro, men belønnet for å bevege seg til nye plasser og for å plassere bomber. I motsetning til agenten uten reward shaping vil den bevege på seg, men med en liten utforsknings-periode vil den ikke nødvendigvis få sjansen til å oppdage alle mulige belønninger. Dersom en belønning knyttet til en handling ikke blir oppdaget, vil dette heller ikke bli en del av de mulige handlingene agenten vil utnytte når utforsknings-perioden er over. Dermed vil den ende opp, i likhet med den forrige agenten, med å gjenta flere av de samme kombinasjonene av handlinger om og om igjen og begrense seg selv til et lite område.

Figur 4

Sammenlignet med andres forsøk på å slå SimpleAgent med reinforcement learning, har resultatet og effekten av vår reward shaping vært relativ lik. I et forsøk på lære en agent med CNN basert modeller ble det oppnådd en gjennomsnittlig belønning på 0,6 og høy suksessrate, men uten reward shaping klarte ikke agenten å lære seg å plassere bomber. Den vil derimot lære seg å unngå bomber, og vil i likhet som vår agent uten reward shaping, vinne om de andre spillerne tilfeldigvis bomber seg selv. [5]

I et annet Pommerman-eksperiment ble det forsøkt å gi belønning for enhver handling, men dette resulterte i at agenten unngikk å plassere bomber da andre handlinger var lettere og tryggere å utføre. Videre ble belønningen for plassering av bomber økt for å gjøre agenten mer aggressiv, men selv om belønningen ble tidoblet, prøvde likevel agenten å finne måter å få høy belønning uten å plassere bomber. [6]

I en masteroppgave ble det implementert en mer detaljert reward shaping, hvor handlinger ble delt inn i ulike kategorier; offensiv, defensiv og mobilitet. Her fikk agenten belønning for mer spesifikke handlinger som å bevege seg i motsatt retning fra bomber, bevege seg mot fiender og hvor man plasserte bomber. Dette bidro til å holde agenten i live lengre enn kjøringer uten reward shaping. [7]

Testing av trente nettverk

Etter trening så ble det kjørt 50 episoder der de ferdigtrente agentene ble brukt mot 3 andre "SimpleAgent" agenter. Målet med dette var å se hvilke handlinger DQN agentene benyttet seg av. Figur 5 viser handlingene agenten tok der det ikke var reward shaping. Man kan tydelig se hvorfor episodene er mye lengre enn de er for agenten med reward shaping. På grunn av at agenten velger og stå stille, altså bruke handlingen "stop", så tar det mye lengre tid før noen av de andre spillere kommer og tar agenten ut.

På figur 6 derimot, kan man tydelig se at agenten går rundt og velger handlinger som vil gi den en bedre belønning. Men også her kan man se at agenten velger noen handlinger mye oftere enn andre, blant annet plassering av bombe som den velger mye oftere enn handlingene som får agenten til å flytte på seg. Reward shaping som ble brukt gir potensielt mye belønning for plassering av bomber og det er tydelig å se at agenten lærte at det er lønnsomt å plassere bomber, til tross for den store risikoen dette medfører.

Figur 5

Figur 6

Konklusjon

Uten reward shaping vil ikke agenten ha noen grunn til å bevege seg rundt på brettet. Den velger heller å vente til de andre agentene dreper hverandre og seg selv. Ettersom den synes det er mer lønnsomt å vinne ved en tilfeldighet og **utnytte** kjente handlinger uten å inngå kompromisser, vil den ikke **utforske** alternative måter å vinne på, selv om det muligens vil gi en høyere belønning i lengden. Dette er et kjent dilemma og å finne balansen mellom utforskning og utnytting er essensen i få gode løsninger innenfor reinforcement learning. [8] Denne oppførselen kan bidra til å få agenten til å leve lengre, ettersom den ikke utsetter seg selv for fare, men det vil ikke nødvendigvis føre til en bedre ytelse da den ikke aktivt gjør noe for å vinne.

For miljøer som Pommerman som er sparsomme med belønninger, kan det være svært vanskelig for en agent å koble sammen et stort antall handlinger mot en mulig belønning langt frem i tid. [9] I slike situasjoner kan reward shaping brukes for å gi en agent en grunn til å utforske andre handlinger slik at den bedre kan lære seg veien mot den fremtidige belønningen. I andre forsøk har agenten vanligvis tilpasset seg en mer defensiv spillestil når reward shaping er tatt i bruk. Den lærer seg å unngå bomber, men vil som regel ikke plassere mange bomber. I motsetning til vår agent, hvor den med reward shaping vil prøve å plassere så mange bomber som mulig og samle opp så mye belønning som mulig med den strategien. Her vil strukturen av belønningen påvirke i stor grad. Reward shaping som er for det meste basert på positive belønninger vil oppfordre agenten til å leve lengre og utnytte lette handlinger for å akkumulere en høyere belønning. På den andre siden, kan en reward shaping som tar i bruk negative rewards oppfordre agenten til å avslutte episoden så fort som mulig i frykt for å samle opp for mange negative belønninger. [10]

Agenten som ble trent med reward shaping har en høyere gjennomsnittlig belønning, men dette betyr ikke nødvendigvis at agenten vinner mer. Den er uten tvil mer aktiv, men den lever også betydelig kortere, da den tar flere risikoer i håp om å få høyere belønning eller bare vil avslutte episoden fortest mulig. I tillegg vil agenten utnytte enkle handlinger som gir lav belønning da dette er lettere tilgjengelig enn å jakte etter fiender. Den kortere levetiden til agenten vil også minske sannsynligheten for at den vinner ved en tilfeldighet som et resultat av at de andre agentene dreper hverandre.

Ut fra resultatene så kan man se at det ikke bare er reward shaping som har noe å si om man får forventet resultat fra agenten eller ikke. I en artikkel skriver OpenAI at de ikke fikk forventet resultat på spillet "CoastRunners" ,der agenten får en belønning for å bli ferdig med kappløpet og fra å samle inn gjenstander på banen. Resultat OpenAI fikk var at agenten gikk i sirkler og samlet inn gjenstander i et lite område. Det førte til at agenten aldri ble ferdig med kappløpet, men fikk høyest poengsum. Deretter skriver OpenAI om andre ting som kunnet hjulpet, for eksempel "human demonstrations" som går ut på å ta opp hva et menneske hadde gjort og videre bruke det som et utgangspunkt når man trener agenten. [11]

Reward shaping kan altså ha store påvirkninger på oppførselen til en DQN agent, altså hvilken handlinger den velger å gjøre, siden agenten vil utforske flere muligheter dersom de gir belønninger. Videre har vi sett på hvordan verdiene i reward shaping og strukturen til reward shaping vil påvirke agenten. Selv om reward shaping er essensielt for oppførselen til agenten kreves det mer enn å bare justere noen verdier for å oppnå en bedre ytelse og høyere suksessrate.

IDATT2502 Dominykas Mazys, Joel Mattias Tømmerbakk, og Minh Dan Nguyen 28.11.2022