10.10.2024

Eirik Tennøfjord

More og Romsdal fylkeskommune

Et bilde som inneholder skjermbilde, tekst, programvare, Fargerikt

Automatisk generert beskrivelse

Q-Learning

Karaktersatt oppgave 1 DTE2602 – Introduksjon Maskinlæring og AI

Innhold

[Terminologi 1](#_Toc179449247)

[Introduksjon 1](#_Toc179449248)

[Teori 2](#_Toc179449249)

# Terminologi

Agent

Miljø

Q-Learning

Monte Carlo

Epsilon-Greedy

Belønning

Q-Matrise

Episoder/Epochs

# Introduksjon

I dagens samfunn har utviklingen innenfor kunstig intelligens (KI) skutt fart. Eksempler på dette kan være chat bot, selvkjørende biler, selvkjørende ferger/båter og sorteringsroboter. Flere av disse punktene dreier seg om navigasjon, som er en sentral del av KI. Navigasjon er en av de grunnleggende utfordringene innenfor robotikk og maskinlæring. Det å kunne navigere i et miljø, for deretter å identifisere hindringer og finne den mest effektive veien kan være avgjørende for flere felt.

En viktig del av det å kunne navigere i et miljø, er forsterkningslæring, der vi bruker en kode (agenten) til å navigere og ta beslutninger basert på miljøet. Q-learning og Monte Carlo er to kjente metoder innenfor forsterkningslæring. Begge disse metodene lærer ved å samle opp belønning over tid. Q-learning lærer ved å kontinuerlig oppdatere en Q-matrise etter hvert som miljøet blir utforsket. Monte Carlo summerer opp belønning og sammenligner disse for å finne den mest effektive ruten.

En annen strategi for å navigere er en metode vi kaller epsilon-greedy. Denne kombinerer q-learning, og monte carlo. Ved å benytte denne metoden, så vil man bruke kjent informasjon om hvilke ruter som gir høyest belønning, samtidig som man vil kunne oppdage potensielt bedre ruter, ved å gjøre tilfeldige handlinger.

For å studere disse forskjellige læringsmetodene er et 2D-kart et egnet miljø. Gjennom bruk av belønningsmatriser og simuleringer kan agenten lære seg å finne effektive ruter fra start punkt til slutt punkt, samtidig som agenten kan manøvrere seg rundt hindringer.

Denne rapporten vil undersøke hvordan man med ulike navigasjonsstrategier, som Monte Carlo og Q-learning kan finne de mest effektive rutene i et 2D-kart. Formålet er kunne vurdere hvordan de ulike metodene fungerer, samt hvor effektive de er. Det er også ønskelig å se på antall episoder (epochs), og hvordan dette spiller inn på de forskjellige strategiene.

Hypotesen i denne rapporten er at en kombinasjon av disse metodene kan gi en optimal løsning for robotens navigasjon. Q-learning vil være mest effektiv hvor man har et stort antall episoder, mens epsilon-greedy kan gi bedre resultater med et mindre antall episoder. Gjennom simulering vil rapporten vise hvordan roboten lærer og forbedrer sine valg over tid, og hvordan den finner de mest effektive veiene fra start til slutt.

# Teori

* Referanse og forklaring til bellmanlikning
* Markov-antakelsen
* Monte Carlo simulering
* Minst én relevant referanse til relevant problematikk knytt til navigasjon

"Teori-delen beskriver teoretiske konsepter som er relevante for rapporten, for eksempel algoritmer, matematiske ligninger, database-modeller, osv. Husk at elementene som brukes må beskrives i teksten (f.eks. hva hver enkelt variabel i en ligning betyr). Husk også referanser."

# Metode

For å utvikle agenten til å navigere i 2D-kartet, ble de benyttet forskjellige metoder innenfor forsterkningslæring. Dette inkluderte: Monte Carlo, Q-learning og Epsilon-Greedy policy. Selve arbeidet ble gjennomført i Python, og ved hjelp av visualiseringsverktøyet "PyGame".

Monte Carlo ble implementert ved at man setter agenten/roboten i start posisjon, deretter fikk den gjøre tilfeldige handlinger som i dette tilfellet var opp, ned, venstre og høyre. Hver gang man kommer til en ny tilstand vil man oppdatere episodens totale belønning.  
Det ble laget ######################################################

Q-Læringen ble implementert ved å initialere en Q-matrise som holdt styr på de forskjellige belønningen for hver tilstand og handling. Agenten startet fra en tilfeldig posisjon i kartet og utfører en handling basert på valgt policy (ϖ).

* Alle valg skal være godt begrunnet og dokumentert med referanse til koden.
* Hele arbeidsprosessen er sammenhengende og dokumentert, ikke unødvendig delt opp.
* Fremgangsmåten er klart og tydelig beskrevet slik at andre kan gjenta ekspermentene.

"Beskrivelse av hvordan arbeidet ble gjennomført i praksis - hvilke verktøy og metoder som ble brukt, implementasjonsdetaljer, osv. Bruk et objektivt språk som fokuserer på hva som ble gjort, og unngå &amp;amp;amp;quot;dagbok-format&amp;amp;amp;quot;. Dokumenter valg som er tatt i koden, og inkluder gjerne kodesnutter der det er relevant, men unngå å liste opp all kode (som skjermbilder eller tekst)."

# Resultat

Resultater som er relevant til problemstillingen er presentert objektivt og oversiktelig

"Presentasjon av resultatene av arbeidet. Formidles gjerne i form av tabeller, grafer, illustrasjoner og/eller bilder i tillegg til tekst. God struktur er viktig for at leseren skal skjønne sammenhengen mellom resultatene og tidligere beskrevet teori og metode. Kommenter gjerne kort i teksten for å fremheve de mest interessante / overraskende resultatene, men pass på å ikke bli for subjektiv."

# Diskusjon og konklusjon

* God selvinnsikt som står til levert kode og oppnådde resultater
* Drøfting av mulige feilkilder, satt i kontekst med framtidig arbeid.

"Diskusjon rundt resultatene: Hvilke resultater er mest interessante? Er resultatene som forventet, eller ikke? Prøv å sette oppnådde resultater i sammenheng med opprinnelig problembeskrivelse/introduksjon. Konklusjonen skal kunne leses ganske frittstående, uten å kjenne alle detaljene i rapporten. Dersom det er aktuelt å fortsette arbeidet, kommenter gjerne hva som bør gjøres videre."