

Øving 8: Reinforcement Learning

TDAT3025 - Anvendt Maskinlæring med Prosjekt

Oktober 2020

Om øvingen

I denne øvingen er oppgave 1 obligatorisk, men dere **trenger kun å velge én av 2A eller 2B**. Alle disse oppgavene har mange gode eksempler tilgjengelig på nett, så det er fullt mulig å ta inspirasjon derifra, men dere forventes å måtte forklare koden under demonstrasjonen for godkjenning. Dersom det er spørsmål eller uklarheter, ikke nøl med å sende en epost til **jonathan.jorgensen@ntnu.no**

1. Q-Læring: Cartpole

Implementer Q-læring og bruk det for å løse cartpole-environmentet.

<https://gym.openai.com/envs/CartPole-v0/>

<https://github.com/openai/gym>

2A. Q-Læring: Gridworld med visualisering

Lag et enkelt *gridworld*-environment. Dette innebærer at environmentet har et diskret rutenett, og at en agent kan bevege seg rundt med fire handlinger (opp, ned, høyre, venstre). Simuleringen terminerer når agenten har nådd et plassert mål-posisjon som gir reward 1. Om man ønsker, kan det legges inn f.eks. solide vegger eller farlige områder som gir straff rundt omkring. Environmentet skal ha samme interface som cartpole (.step(a)-funksjon, og .reset())

Deretter skal implementasjonen av Q-læring fra forrige oppgave brukes for å trene en agent i environmentet. Til slutt skal Q-verdiene visualiseres inne i selve environmentet, og dette kan gjøres på flere måter. En måte er å fargelegge rutene basert på den høyeste Q-verdien fra tilsvarende rad i Q-tabellen. Alternativt så kan man tegne inn piler som peker i samme retning som handlingen med høyest Q-verdi.

Tips: Biblioteket *pygame* er veldig greit for å lage visualisering av environmentet.

2B. DQN: Cartpole

Implementer en enkel versjon av DQN for å lære cartpole. I praksis betyr dette at tabellen fra første oppgave skal erstattes av et nevral nettverk, og at det må lages et *replay buffer*. DQN er en algoritme som har fått mange ”utvidelser” over tid, men ingen av disse skal være nødvendige å implementere for å løse oppgaven effektivt.