IT3105 – Kunstig intelligens programmering

Prosjekt 1: Texas Hold’em Poker

av

Ole-Petter Olsen

Lars Solvoll Tønder

1. Kodestruktur

Del 1: Poker simulator

Her er alt samlet i én fil; *Poker.py*. Den eneste klassen er *Player*, som inneholder informasjon om hver enkelt spiller, samt metoder for å ta hånd om handlinger en player kan gjøre, samt en metode, *Assess()*, som avgjør hvilken handling spilleren skal ta basert på hvilken type spiller det er. Her vil spillere fra hver av de 3 fasene oppføre seg forskjellig, samt at spillerne kan ha ulike personligheter innad i fase-typen. Selve kjøringen av runs tas hånd om av main-metoden i *Poker.py*. En kjøring gjøres ved å dele ut hullkort til alle spillerne etterfulgt av *InitialBet()*, trekking av flop-kort, *FlopBet()*, trekking av turn-kort, *TurnBet()*, trekking av river-kort, *RiverBet()* og til slutt *Showdown()*, hvis det er flere spillere igjen i spillet.

Del 2: Hand strength

For del 2 av prosjektet er koden delt inn i 2 filer; *hand\_strength.py* og *pre\_flop.py*. *hand\_strength.py* har metoden *calculateHandStrength(hand, numberOfOpponents,tableCards)* som regner ut håndstyrke for et gitt par med hullkort ved å regne ut antall seiere, uavgjort og tap mot alle andre mulige par med hullkort og 3,4 eller 5 delte kort. Formelen for håndstyrke er som følgende:

Der k er antall motstandere.

*calculateHandStrength()* bruker en hjelpefunksjon *calculateOutcome(player, opponent)* som returnerer 1, 0 eller -1, avhengig av om player vinner, spiller uavgjort eller taper mot opponent.

Del 2: Pre-flop rollout

*hand\_strength.py* inneholder også metoden *getPreFlopTable()* som returnerer en 4-dimensjonal tabell med sansynlighetene for å vinne for alle hullkortkombinasjoner. Den henter verdiene fra en fil som har følgende format:

<int players>  
<string suited>  
2x2  
3x2 3x3  
4x2 4x3 4x4  
.  
.  
.  
14x2 14x3 ... 14x14  
<int players>  
<string suited>  
2x2  
.  
.  
.  
14x2 ... 14x14

De første 9 tabellene har players fra 0 til 8 og suited = ”unsuited”. De neste 9 tabellene har players fra 0 til 8 og suited = ”suited”.

Sannsynlighetene kan hentes ut av tabellen på følgende måte:

pre\_flop\_table[<høyeste verdi> - 2][<laveste verdi> - 2][<suited/unsuited>][<antall motstandere> - 1],

der høyeste og laveste verdi er et tall mellom 2 og 14 og antall motstandere et tall mellom 1 og 9. Dersom man har 2 kort med samme farge skal man bruke 1 for <suited/unsuited> og 0 ellers.

Metoden som lager filen ligger i *pre\_flop.py*. Den starter med å lage alle mulige kombinasjoner med hullkortpar og kjører 5000 rollouts for hver kombinasjon og lagrer den beregnede sannsynligheten for å vinne i en tabell på samme format som nevnt ovenfor (pre\_flop\_table). Det gjøres i metoden *calculateProbabilities()*. Til slutt skrives alle sannsynlighetene ut til en fil av samme formatet som er nevnt ovenfor. Dette gjøres i metoden *makePreFlopFile()*.

Del 3: Opponent modelling

Skriv noe her

2. Logikk for vedding

Del 1

Del2

Del3

3. Opponent models

4. Resultater fra kjøringer

Del 1

Del 2

Del 3

5. Diskusjon rundt strategier

Del 1

Del 2

Del 3