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

I fase 3 skulle vi bruke modellering av motstanderen til å bestemme adferden til pokerspillerne våre. Vi startet derfor med å lage metoder som logget spillerenes trekk for hver budrunde og som lagret disse verdiene, samt miljøet de ble utført i, i en tabell vi valgte å kalle estimatorTable. Miljøet et trekk ble utført i består av følgende verdier:

* Kontekst: En tallverdi utregnet fra forskjellige variabler
* Hullkort: Kortene spilleren hadde på hånden den runde
* Delte kort: Kortene som lå på bordet
* Håndstyrke: En tallverdi utregnet som vist tidligere for å beregne hvor sterk en hånd er

Da vi skulle bestemme hvilke variabler som skulle gå til å regne ut kontekstverdien startet vi først med ganske mange verdier som blant annet: Antall gjenværende spillere, hvor mange ganger noen har ”raiset” i løpet av runden, hvilken runde man er i, pot odds og hvor mye penger motstanderene har. Det viste seg derimot raskt at det ble alt for mange kontekstverdier. Dette førte til at fase 3 spillerene fikk et alt for lite grunnlag til å basere sine egne handlinger på, og vi måtte derfor korte ned på antall kontekstverdier. Da ingen av oss på gruppen har noe særlig erfaring fra poker måtte det noe testing til før vi kunne bestemme oss for hvilke variabler vi skulle basere verdien på. Det hele endte med at vi valgte å basere kontekstverdien på bare 3 forskjellige variabler, henholdsvis: hvilken runde man er i, antall gjenværende spillere og hvor mye en spiller har lag til potten denne runden. Disse verdiene egnet seg godt til diskretisering, samt at de kunne brukes til å gi et helt greit bilde av motstanderenes handlinger, uten å måtte ta ibruk alt for mange variabler.

Fase 3 spillerene kunne dermed regne ut den gjennomsnittlige hånden en gitt spiller hadde under en viss kontekst da han gjorde forskjellige trekk, noe som ble implementert gjennom metoden GuessHand(). Da vi hadde 3 forskjellige personlighetstyper valgte vi at hver personlighetstype skulle bruke disse verdiene på forskjellige måter for å velge hvilken handling de selv skulle gjøre. Disse taktikkene for handlingsmønster diskuteres i det kommende kapittel.

2. Logikk for vedding

For hver fase av prosjektet brukte vi tre forskjellige personlighetstyper til å definere adferden til de spillerene. Vi valgte en forsiktig, pågående og ”bløffende” type, som hver og en har forskjellige adferdsmønstre. De forsiktige spillerene vil generelt bare spille dersom de føler at de har en god hånd, eller dersom det ikke koster noe å være med videre. De pågående spillerene er derimot litt mindre forsiktig, og vil generelt sett fortsette å være med i runden selv om det kommer til å koste. Til sist har vi de bløffende spillerene, som kan tenke seg til å spille med en hvilken som helst hånd, bare den ikke er alt for stor sannsynlighet for å tape.

Fase 1

Under fase 1 er det helt tilfeldig hvilken handling spillerene utfører under den innledende budrunden før floppen. Hvilken handling de tar for seg i de neste rundene kommer bare an på håndstyrken deres.

* Konservativ spiller: En konservativ spiller vil bare ”raise” dersom han har en bedre hånd enn et par og dersom han samtidig ikke har brukt noe særlig penger denne runden. Han vil derimot fortsatt være med så lenge han slipper å betale for det, eller dersom han har en helt grei hånd og enda så og si ikke har brukt noe penger denne runden
* Pågående spiller: En pågående spiller vil som en konservativ spiller her også bare ”raise” dersom han har en bedre hånd enn et par, men han bryr seg fint lite om hvor mye penger han har brukt denne runden. Det samme gjelder dersom han har en helt grei hånd, men han er her ikke like villig til å betale så alt for mye for å fortsatt være med i budrunden.
* Bløffende spiller: En bløffende spiller vil ”raise” med en hvilken som helst hånd så lenge han ikke har brukt for mye penger denne runden. Dersom han har brukt for mye penger og fortsatt må betale for å være med i budrunden vil den bløffende spilleren heller kaste hånden enn å risikere å miste alt for mye.

.

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