**Rapport sommer 2020**

Er ikke helt sikker hvad jeg skal skrive her i rapporten, har ikke riktig hatt noen overordnet problemstilling (jeg burde sikkert ha vert bedre til å spørre direkte etter en). Har laget en liste over problemstillinger jeg har jobbet med og hvordan de er løst for alex og bandit miljøene under. Hvis dere vil ha noen av plottene eller trente modeller er det bare å si ifra :)

**Alex environment (fil - “a2c\_meta-alex.ipynb”):**

**Problemstilling 1:**

Bygge en funksjon som beregner sannsynlighetene for at et dist\_env er i en bestemt tilstann.

**Løsning:**

Løst med funksjonene “get\_probabilities” og “action\_values” i dokumentet. “get\_probabilities” gir sannsynlighetene for en gitt tilstann mens “action\_values” går gjennom et miljø der en agent har handlet og returnerer sannsynligheter, forventningsverdier og bedste action til hvert tidssteg. Disse er blitt innebygd i Benchmark klassen slik at benchmark.episodes har en “best” key som gir den beste action til hvert tidssteg. Bemærk at forventningsverdien er kun tager højde for belønningen av handlingen og ikke hvilken informasjon som fås ved å svare rett/galt eller vente.

**Problemstilling 2:**

Undersøke hvordan forskjellige agenter oppfører seg i miljø med forskjellige forhold mellom belønning og straff ved rett/galt svar.

**Løsning:**

Trene agenter med forskjellige forhold mellom belønning og straff, og igjen evaluere alle agentene ved hjelp av Benchmark klassen i miljø med forskjellige forhold. Agentene ble trent og evaluert i miljø med belønning på hhv 1, 2, 3, 4, 5, 6, 7, 8, 9 og 10. Der var ingen straff for å vente, og straffen for å gjette feil var fast på 10. Det ble gjort plot av den gjennomsnittlige belønning, andelen av “beste” actions samt “vente-raten” (forholdet mellom antallet av “vente-actions” og “ikke-vente-actions”).

**Resultat og diskusjon:**

“vente-raten” så ikke ut til å endre seg med økt belønning i miljøet der agenten handlet. Agenter trent i miljø med belønning mindre enn 8 gjorde stort set ikke annet enn å vente uavhengig av miljø de ble evaluert i. Det så umiddelbart ikke ut som at det var noen nevneverdig forskjell på de resterende agenter i hvordan de handlet. Grafene er ikke med her da de ikke er så veldig interessante. Det hadde kanskje vært smart å gi agentene en straff for å vente for å få de til å handle oftere?

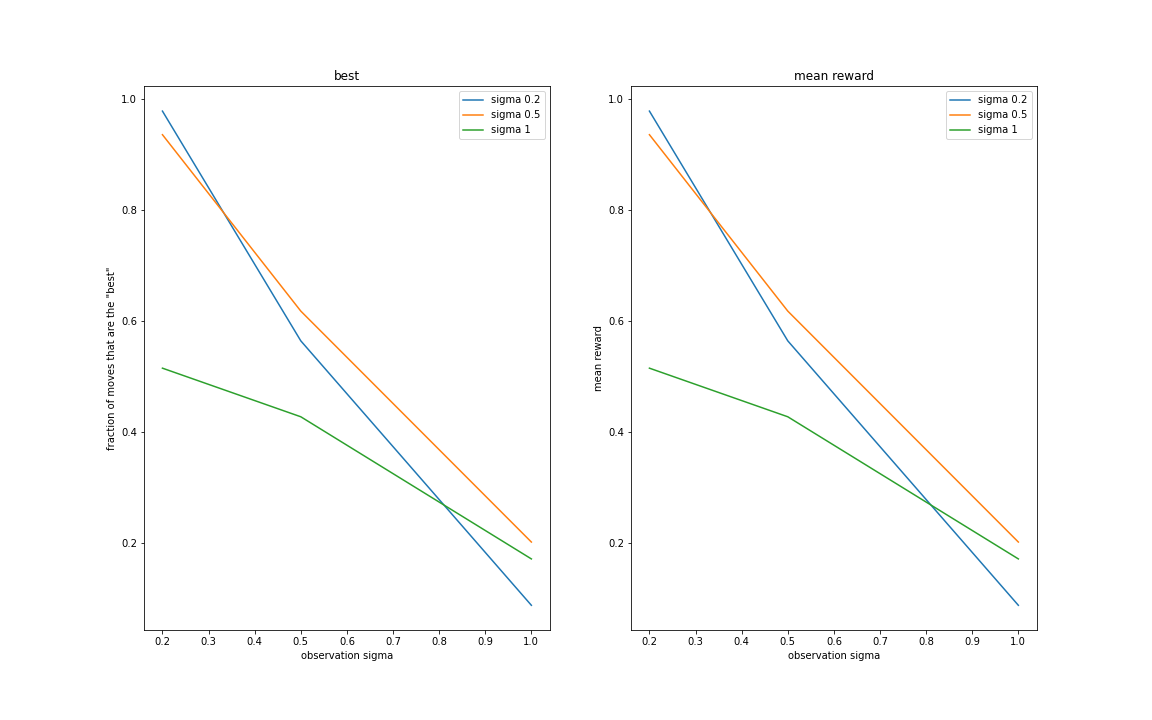
**Problemstilling 3:**

Undersøke hvordan agenter med forskjellige observation-sigmaer handler i forskjellige miljø.

**Løsning:**

Her ble agenter trent og evaluert i miljø med observation\_sigma på: 0.2, 0.5 og 1. Gjennenmsnittlig belønning og andelen av ‘beste’ actions ble plottet med observation\_sigma på x-aksen.

**Resultat og diskusjon:**

****

Ser at det er vanskeligere å gjette rett ved større observation sigma hvilket er forventeligt. Ser også at agenten trent med observation\_sigma = 0.5 klarer å slå agenten trent med observation\_sigma = 1 i miljøet der observation\_sigma = 1. Her var det kanskje litt dumt at jeg ikke plotter “vente-raten” da jeg mistenker at de er forskjellige for alle tre agenter (men ikke nødvendigvis avhenger av miljø de blir evaluert i). I tillegg kunne det kanskje være interessant å se på flere verdier av observation\_sigma’er.

**Bandit env (fil - a2c\_meta-bandits.ipynb):**

**Problemstilling 1:**

Prøve å gjenskabe dele av “wang et al 2017 - Learning to reinforcement learn” der de klarer å trene agenter på alle vanskelighetsgrader og slå thompson-algoritmen på “dependendt\_bandit”.

**Løsning og diskusjon:**

Denne er ikke løst endda. Det erskrevet en thompson agent, til bruk i sammenlikning. I tillegg er det implementert Ray[tune] som skal kunne trene flere agenter over mange cpu’er samtidig, og hjelpe med å optimere valget av hyperparametre. Det kan også være det er en bug her et sted. Tune har forskjellige måter og implementere tidlig stopping ved trening av nettverk samt metoder for å velge beste hyperparametre, disse forklares litt mer i koden og er sikkert smart og leke litt med.

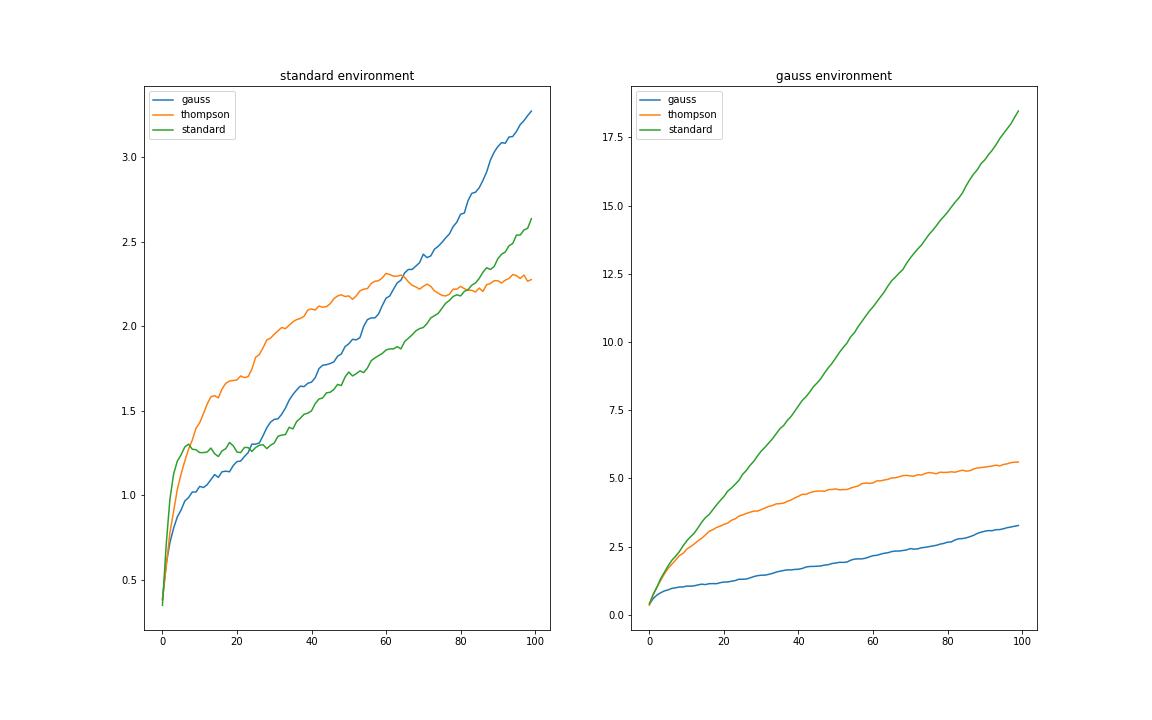
**Problemstilling 2:**

Undersøke hvordan agenter trent i 2 forskjellige typer bandit environments handler på tvers av miljø.

**Løsning:**

Lage en ny type bandit, “GaussBandit” som gir en belønning normaltfordelt om en verdi. Denne bandit har samme forventningsverdier som de andre bandits hadde fra før, altså 0.9 og 0.1 for “easy” etc. Deretter trene nettverk på en GaussBandit, og evaluere denne og et nettverk trent på en standard bandit i begge miljø. I tillegg er det skrevet en thompson agent til bruk på en GaussBandit for sammenlikning. Sammenlikningen gjøres igjen ved å plotte kumulativ regret for agenter i et ‘easy’ environment.

**Resultat og diskusjon:**

****

Forventningsverdien til den kumulative regret i et “easy” environment er 40 for en agent som handler tilfeldig. Ser derfor at begge agenter handler bedre enn tilfeldig utenfor miljøet der de er trent. Det er også interessant å se hvor lite forskjellen er mellom de to nettverk i et standard miljø. Det hadde vert interessant også å undersøke andre vanskelighetsgrader og varianser i gaussmiljøet.