Oppsummering

Bjørn Kristian Punsvik

Del 1

De første ukene gikk mest til oppsett av det vi trengte for å utføre prosjektet. Dette er ting som dependencies til tensorflow, TF-Agents og fikling slik at miljøet skulle kunne skjøre og trenes i det hele tatt. Dette var uten maskinlæringsdelen og bare med random input som action uten noen form for læring. I starten var dette et problem fordi vi ikke fant andre som hadde løst samme miljø med kombinasjonen Tensorflow 2.0 og Tf-Agent. Etter hvert fant vi ut noe som fungerte og vi hadde da noe å bygge ut ifra. Det ble satt opp Overleaf for å kunne samskrive latex til rapporten slik at vi kunne smått begynne på det ved siden av prosjektet. Det ble også satt opp et Git repo på min personlige server slik at samskrivingen av kode skulle gå smertefritt. Dette viste seg å være spesiellt godt å ha når vi senere fikk tilgang til en VM for trening.

Når ting begynte å fungere begynte lesingen om OPEN AI og PyBullet. Leste alt som var av offentlig dokumentasjon som omhandlet disse miljøene og RL. Så og leste også artikkler om RL på youtube og blogposter for å få en bredere forståelse av RL i sin helhet siden Jonathans leksjon bare var en introduksjon og jeg følte jeg ikke kunne nok. Hele denne tiden hadde jeg problemer med at min innstallasjon av Debian 18 på laptopen ikke gjennkjente at tensorflow 2.0 var kommet ut og tilgjengelig for installasjon og dermed nektet å innstallere det. Dette skulle være tilfellet under hele prosjektet. Debian 19 på den stationære kontorPC'en plasert på kontoret til TIHLDE Drift var villig til å installere og kjøre Tensorflow 2.0 kode, men selv om den ikke taklet å kjøre kode på GPU fikset den alt på CPU. Alt var satt opp riktig (tf-gpu, cuda, drivere, osv.), men hardware-en var nok for slitt til å kjøre noe så nytt selv om den var over cuda 3.5 sertifisert. Dette skulle vise seg å være unødvendig siden vi etter hvert fikk tilgang til VM.

Første veiledningmøte gikk mest ut på å forklare problemstillingen til Jonathan og få en guide på at det er snakk om policy gradient og at dette er et vanskelig problem som har continius action space i RL som kan gjøre at vi vil ha problemer med å løse miljøet. i tilegg ble vi advart mot å bruke Open AI sin MuJoCo gym siden den hadde stygge lisenser tilknyttet seg. Vi skulle heller bruke noe fra PyBullet som var gratis. Vi ble anbefalt å gjøre mer

research og lese det andre har gjort og fortalt at prosjektet kunne inneholde en del kok, men at det var forståelsen som var det viktigste.

Det ble deretter en del research på policy gradient-er, hovedsaklig PPO, og en del 'talks' fra tensorflow og teamet bak TF-Agent om RL og koblingen mellom disse.

Del 2

Etter at halve tiden var gått og C++ faget og obligatoriske oppgaver innen linjeforeningen var unnagjort ble det mer tid til prosjektet. Det startet med at vi fikk vite at miljøet ikke trengte å bli løst, så lenge arbeidet og rapporten var god. Det ble snakket om hva vi skulle foreta oss videre for vi var begge usikkre på hva neste steg var. Det kom også frem problemene vi hadde med treningen på våre egne maskiner og ble tilbudt en VM fra NTNU LABen. Jeg skrev en rask mail til Ole som han videresendte til LABen og vi fikk tilgang til en maskin dagen etter.

Det neste som måtte gjøres var å sette opp denne slik at vi kunne trene på den. Det var derfor viktig at dette ble gjort ordenlig med virituellt python miljø og de programmer som trengtes for å overvåke, kjøre og evaluere treningen. Det nærmet seg helg og begge prøver å holde studietiden til arbeidstid på hverdagene så vi viste at helgen ville være en tørke fra fredags ellermiddag til mandags morgen. Vi bestemte oss da for å bruke fredagen til å utvikle scrip og tilpasse koden med flagg slik at vi kunne prøve ut mange forskjellige permutasjoner av hyperparametere for å se hvilket som kunne fungere. Dette skulle vise seg å være starten på hvordan prosjektet skulle ende opp. I løpet av den helgen ble det kjørt og loggført mye data vi kunne analysere kommende mandag.

Neste veiledningmøte gikk ut på at vi analyserte grafene i tensorboard til å ikke gi noen gode resultat. Vi vurderte derfor å bytte algoritme fra PPO til SAC om vi fant god måte å gjøre det på. Det skjedde aldri. Vi ble også anbefalt å finne mening i grafene og rådet til å starte på rapporten. Dagen gikk stort sett ut på å sette opp riktig utforming for referanser i latex slik at det kom i NTNUs ønskede stil; Harward, og litt skriving på rapporten av det vi viste til nå.

Den neste tiden gikk til å tilpasse et script som kunne rendre modeller i real time eller video slik at vi kunne se hva som skjedde med mauren og prøve ut forskjellige checkpoints av samme modell slik at vi så fremgangen, om der var noen. Jeg satt også opp andre miljø fra PyBullet og trente disse for å

bli mer kjennt med miljøene i tilfellet det var noe vi gjorde fundamentalt galt. Cartpole problemet ble skrevet, trent og løst enkelt. Minotaur ble også skrevet og trent, men prøvde der å bruke SAC for å se om dette var noe bedre, men minotauren ble aldri ferdig trent og det var tilbake til mauren uten noen åpenbaring på hva vi kunne ha oversett. Også litt rapportskriving.

På neste og siste veiledningsmøte ble det bestemt at vårt prosjekt var bedre som en grid-search og ikke en løsning av ant. for det vi i bakgrunnen hadde gjort hele tiden var meta-learning og søk på hvilke hyperparametere som kunne fungere best. Dette var da bare et spørsmål å endre noen settninger i rapporten og fokusere på hyperparameterene fra nå av. Dette var en potensiellt mye mer spennende vinkling siden det sikker var få papirer om dette temaet og om noen løste ant for 14 gang var det ikke noe nytt og spennende. Vår research fant bare ett papir som lignet på det vi nå gjorde. Resten av tiden gikk bare til skriving av rapport og rensing av kode slik at den var leselig og fin til innlevering

Log

Torsdag 17. okt : Valgt prosjektoppgave. Satt opp git på personlig server. Satt opp Overleaf-prosjekt for latex-samskriving av rapporten.

Tirsdag 22. okt : Lest PyBullet Quickstart Guide om RL i tensorflow. youtube: om RL, introduksjon osv. Deep Reinforcement Learning: Pong from Pixels - Andrej Karpathy blog. innstalled dependencies.

Tirsdag 29. okt : Veiledningsmøte - fokuser på policy gradient. Dette er et continius actionspace i RL. Gjør mer research, les det andre har gjort. Kjør masse cloner fra github. Forståelse er viktig.

yt: Policy Gradient methods and Proximal Policy Optimization (PPO): diving into Deep RL!
yt: Reinforcement Learning in TensorFlow with TF-Agents (TF Dev Summit '19).
https://www.youtube.com/watch?v=-TTziY7EmUA **Tirsdag 5. nov :** Veiledningsmøte - Miljøet er vanskelig, Det er ikke å løse den som er viktigst. En god rapport kan få en bra karakter selv om løsningen ikke er nådd. Snakket om en VM som kan brukes til treningen siden maskinene våre varf or treg og GPU akselerert maskin ikke ville gjøre stort forskjell.

Sendt mail til Ole om VM, virderesendt, fikk svar dagen etter.

Onsdag 6. nov : Satt opp training VM, venv, dependancies,Tensorflow,

tensorboard, scripts for kjøring.

Fredag 8. nov : Satt opp helgescripts som skulle alle permutasjoner av forskjellige variabler ila. helgen.

Tirsdag 12. nov : Veiledningsmøte - Prøv ut sac, let etter mening fra grafene, skriv rapport.

Satt opp rammeverk for refferanser i latexdokumentet. skreiv litt på rapporten.

Torsdag 14. nov : Prøvde å få RenderAnt.py til å rendre på gpuen på kontoret og finne en modell som kunne ta et skritt.

Fredag. 15. nov : Satt opp cartpole og trente den ferdig, prøvde minotaur sac uten hell.

Mandag 18. nov : skrevet på rapporten, mest på introduksjon.

Tirsdag 19. nov : Veiledningsmøte - Endre rettning til grid-search og ikke løsing av ant. rapporttilbakemeldinger

skrevet rapport. ryddet i kode.

Onsdag 20. nov : skrevet rapport.

Torsdag 21. nov : skrevet rapport. skrevet oppsummering. klargjort kode for levering.