Александар Миланов
Агентно-базирани системи

ОПТИМИЗАЦИЈА НА ПАРАМЕТРИТЕ НА АГЕНТИ ЗА “ЗАСИЛЕНО УЧЕЊЕ” ЗА ACROBOT

Во овој труд се презентираат експериментални резултати од оптимизација на параметрите на два агенти за засилено учење: DQN и PPO, применети на околината Acrobot-v1. Низ систематски пристап на пребарување на мрежа на параметри, беа тестирани различни комбинации на хиперпараметри со цел да се постигнат најдобрите можни перформанси на агентите. Резултатите покажаа значителни разлики во перформансите помеѓу различните конфигурации, при што PPO ги надмина перформансите на DQN во речиси сите сценарија.

I. Вовед

Засиленото учење (Reinforcement Learning) [RL] е поддисциплина на машинското учење каде што агентите учат оптимални политики преку интеракција со околината. Задачата на RL агенти е да преземат последователни одлуки во динамична околина, со цел да ги максимизираат наградите. Една од најважните задачи при дизајн на RL агенти е изборот на соодветни хиперпараметри, како што се стапката на учење и големината на партиите, кои имаат директно влијание врз перформансите на моделот. Овој труд се фокусира на агентите DQN и PPO, применети на околината Acrobot-v1, и врши анализа на перформансите низ различни комбинации на хиперпараметри.

Целта на ова истражување е да се оптимизираат хиперпараметрите на DQN и PPO агентите со цел да се постигнат највисоки можни резултати. Задачата на Acrobot е да се контролира двоен зглобен робот за да се постигне висина на зглобот што е над одредена цел.

II. Сродни истражувања

Многу истражувачи го имаат разгледувано проблемот на оптимизација на RL агенти за задачи како што е Acrobot. Во трудот на Mnih et al. (2015), предложен е Deep Q-Learning (DQN), кој покажа успешни резултати на различни контролни задачи. Слично на тоа, Schulman et al. (2017) предложија Proximal Policy Optimization (PPO), кој комбинира стабилност и ефикасност при учење. Најновите истражувања се насочени кон комбинации на различни техники како Double DQN и PPO со адаптивни параметри.

III. Опис на агентот, методите и експерименталната поставеност

Овој проект користи два RL алгоритми: **Deep Q-Networks (DQN)** и **Proximal Policy Optimization (PPO)**, имплементирани во библиотеката *Stable Baselines 3*. Експериментите се спроведени на околината Acrobot-v1, која е класична контролна задача каде агентот мора да научи да лансира зглобен робот за да ја постигне целта.

Методологија

1. **DQN**: Алгоритам кој користи Q-табели заменети со длабоки невронски мрежи за евалуација на акции.

2. **PPO**: Политички градиентен алгоритам кој стабилизира учењето преку прилагодени граници за ажурирање на политиката.

Параметри на експериментите

1. **Стапка на учење (Learning Rate)**: Го контролира чекорот со кој агентот ги ажурира тежините на моделот. Преголема стапка на учење може да доведе до дестабилизирање на процесот на учење (агентот може да пропушти важни информации), додека прениска стапка на учење може да го направи процесот премногу бавен и да не доведе до конвергенција. Вредности помеѓу 0.0001 и 0.1 се типично користени за RL задачи, а нашите експерименти покажаа дека стапка во рангот од 0.05 до 0.001 е особено успешна и за DQN, и за PPO.

2. **Дисконтен фактор (Gamma) [γ]**: Одредува како агентот вреднува награди во иднината. Вредности на γ блиску до 1 значат дека агентот ќе ги вреднува идните награди повеќе, додека помали вредности ја нагласуваат краткорочната награда. За многу задачи како Acrobot, долгорочната награда е важна, затоа вредностите од 0.95 до 0.99 се често користени. Ако γ е прениска, агентот може да биде фокусиран само на непосредни награди, додека ако е превисока, агентот можеби ќе игнорира краткорочни награди, што е несоодветно за некои динамични задачи.

3. **Големина на партии (Batch Size)**: Се користи за да се одреди колку примероци агентот ќе користи пред да ги ажурира тежините. Поголемите партии ја стабилизираат тренингот со помалку флуктуации, но бараат повеќе ресурси и време за обработка. Помалите партии овозможуваат побрзо учење, но може да предизвикаат нестабилност. За алгоритмите DQN и PPO, испробани се партии од 32 до 256, каде што големините 32 и 64 се покажаа како оптимална рамнотежа помеѓу брзината на учење и стабилноста на моделот.

IV. Резултати

Евалуацијата на агенти беше спроведена преку мерење на **средната награда (mean reward)** низ 10 епизоди со по 10,000 чекори. Во прилог е табела со резултати од три тестирања на кодот за барање на оптимални параметри:

Алгоритам	Learning Rate	Gamma	Batch Size	Mean Reward
DQN	0.01	0.98	64	-109.2
DQN	0.001	0.96	32	-127.7
DQN	0.05	0.98	32	-104.5
PPO	0.01	0.97	64	-85.1
PPO	0.05	0.99	32	-84.0
PPO	0.001	0.99	32	-76.7

Анализа на резултатите

Како што се гледа од резултатите, PPO агентот постигнува значително подобри резултати од DQN агентот, со најдобра средна награда од -76.7. Во контраст, DQN постигнува најдобра награда од -104.5. Ова покажува дека PPO подобро ги обработува дискретните акции во Acrobot-v1 околината. Истото се забележува и по извршување на кодот за визуелизација (со стандардни параметри), каде повторно PPO се истакнува како подобар.

Ако зголемиме вредноста на чекорите за учење од 10,000 до >50,000 би овозможиле подлабока оптимизација на агентите, но значително би зголемиле времето на тренирање.

И покрај релативно ниската награда во споредба со слични истражувања, резултатите се во линија со очекувањата. PPO обично се користи за задачи со повеќе комплицирани динамики и подолгорочни награди, каде што дисконтираниот фактор има поголема улога. Во овој случај, малите вредности на стапката за учење дозволуваат попрецизно учење на политиките.

V. Заклучок

Во овој труд демонстриравме дека PPO алгоритмот е посоодветен за решавање на Acrobot-v1 проблемот во споредба со DQN преку систематско пребарување на хиперпараметри. Со натамошни подобрувања, како адаптивни алгоритми за хиперпараметри, можно е да се постигне уште подобра ефикасност. Во идни истражувања, се препорачува разгледување на различни структури на невронски мрежи и тестирање на агенти со повеќе чекори на учење, како и експериментирање со други RL алгоритми кои се попогодни за задачи со сложена динамика.