МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

Батбаярын Бямбаням

DDPG бататгасан сургалтын үйлдлийн шуугианыг турших, харьцуулах

(Effect of action space noise for DDPG RL algorithm)

Мэдээллийн технологи (D061304)
Бакалаврын судалгааны ажил

Улаанбаатар

2021 оны 02 сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

DDPG бататгасан сургалтын үйлдлийн шуугианыг турших, харьцуулах

(Effect of action space noise for DDPG RL algorithm)

Мэдээллийн технологи (D061304)
Бакалаврын судалгааны ажил

Удирдагч: _____ Г.Гантулга

Хамтран удирдагч: _____

Гүйцэтгэсэн: _____ Б.Бямбаням (17B1NUM1479)

Улаанбаатар

2021 оны 02 сар

Зохиогчийн баталгаа

Миний бие Батбаярын Бямбаням ”DDPG бататгасан сургалтын үйлдлийн шуугианыг турших, харьцуулах ” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

ЗУРГИЙН ЖАГСААЛТ

ХҮСНЭГТИЙН ЖАГСААЛТ

Кодын жагсаалт

УДИРТГАЛ

Reinforcement learning буюу бататгасан сургалтад тасралттай үйлдлийн хувьд суралцах үйл явц нь санамсаргүй үйлдлийг сонгох замаар явагддаг. Харин үргэлжилсэн үйлдлийн хувьд суралцах үйл явц нь үйлдэлд шуугианыг нэмэх замаар явагддаг. Deep Deterministic Policy Gradient (DDPG) бол тасралтгүй, үргэлжилсэн үйлдлүүдийг сурахад зориулагдсан алгоритм тул action space noise буюу үйлдлийн шуугиан ашиглагдана. Энэ судалгааны ажлаар энэхүү үйлдлийн шуугианыг туршин, харьцуулах бөгөөд ямар үр нөлөөтэй болохыг тодорхойлоно.

0.1 Зорилго

DDPG бататгасан сургалтын үйлдлийн шуугианыг туршин, харьцуулж энэ шуугиан моделийг сургах үйл явцад хэрхэн нөлөөлж буйг харах, дүгнэлт гаргах

0.2 Зорилтууд

-
-

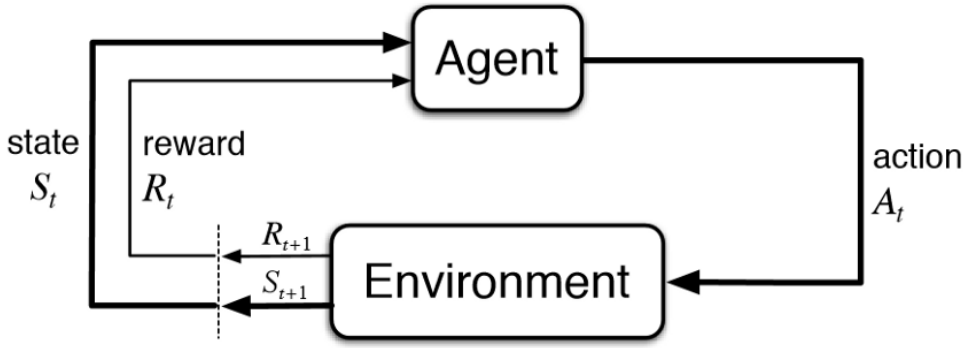
1. СУДАЛГАА

1.1 DDPG алгоритм

Алгоритмыг тайлбарлахаас өмнө Reinforcement Learning буюу бататгасан сургалтын талаар бага зэрэг тайлбарлая. Цаашдаа RL гэж товчилон бичинэ.

1.1.1 Бататгасан сургалтын алгоритм

RL нь агент болон орчин гэсэн хоёр хэсгээс тогтдог. Орчин гэдэг нь агент ажиллаж байгаа объектыг, агент гэдэг нь RL алгоритмыг илэрхийлнэ.



Зураг 1.1: Бататгасан сургалтын бүтэц

Орчин нь агент руу төлөвийг илгээх байдлаар эхлэдэг бөгөөд агент нь мэдлэг дээрээ тулгуурлан тухайн нөхцөл байдалд хариу үйлдэл үзүүлэнэ. Үүний дараа орчин агент руу дараагийн төлөв болон reward-ыг илгээнэ. Агент нь орчиноос хүлээн авсан reward-аар мэдлэгээ шинэчлэнэ. Энэ давталт нь орчиноос дуусгах төлөв илгээх хүртэл үргэлжилнэ. Ихэнх RL алгоритмууд дээрх байдлаар ажилладаг.

1.1.2 Бататгасан сургалттай холбоотай нэр томъёо

- Action (A): Агентийн хийх боломжтой бүр алхамууд
- State (S): Орчиноос буцаж ирэх тухайн нөхцөл байдал
- Reward (R): өмнөх үйлдлийн үр дүнд гарсан ололт
- Policy (π): Агент одоогийн төлөв байдалд үндэслэн дараагийн үйлдлийг тодорхойлоход ашигладаг стратеги.
- Value (V): урт хугацааны ололт
- Q-value эсвэл action-value(Q): Value-тай төстэй. Гэхдээ одоогийн үйлдэлийг нэмэлт параметрээр авдаг.

Model-free болон Model-based

Model нь орчны динамик загварчлалыг илэрхийлдэг. Загвар нь шилжилтийн магадлалыг T(s1 | (s0, a)) одоогийн төлөв s0 ба үйлдэл a хоёроос сурч, дараагийн s1 төлөвт шилждэг.

Model-free гэдэг нь мэдлэгээ шинэчлэхийн тулд туршилт ба алдаанд тулгуурладаг. Төлөвүүд болон үйлдлүүдийг хадгалах шаардлагагүй.

On-policy болон off-policy

On-policy агент нь value-г одоогийн policy-г ашигласан одоогийн үйлдэлд тулгуурлан сурдаг. Харин off-policy агент өөр нэг policy-г ашигласан үйлдэл a*-д тулгуурлан сурдаг.

1.1.3 DDPG алгоритм

Deep Deterministic Policy Gradient (DDPG) бол үргэлжилсэн, тасралтгүй үйлдлүүдийг сурахад зориулагдсан model-free off-policy алгоритм юм. Q-функц ба policy-ыг зэрэг сурдаг алгоритм

юм. Q-функцыг сурахын тулд off-policy өгөгдөл болон Bellman тэгшитгэлийг ашигладаг. Мөн policy-г сурахын тулд Q-функцыг ашигладаг.

DDPG алгоритм дараах 4 неороны сүлжээг ашигладаг:

- θ^Q:Q-network
- θ^μ:Deterministic policy function
- $\theta^{Q'}$:target Q network
- $\theta^{\mu'}$:target policy network

Q-network болон policy network хоёр нь Actor-critic аргатай маш төстөй.

- Actor (Deterministic policy function) - төлөвөөс хамааран үйлдлийг санал болгоно. Төлөвийг оролтоор авч үйлдлийг гаргана.
- Critic (Q-network) - төлөвөөс хамаарсан үйлдэл нь сайн эсвэл муу болохын урьдчилан таамагладаг. Төлөв болон үйлдлийг оролтоор авч Q-value-г гаргадаг.

Target network нь суралцсан сүлжээнүүдийг хянаж байдаг эх сүлжээнүүдийнхээ цагийн хоцрогдолтой хуулбарууд юм. Эдгээр сүлжээг ашиглан тогтвортой сурах байдлыг сайжруулдаг.

Доорх зурагт DPDG алгоритмын pseudo-code-ыг харууллаа. Үүнийг 4 хэсэгт задлан тайлбарлаж болно.

- Туршлагаа хадгалах (Experience replay)
- Actor болон critic сүлжээг шинэчлэх
- Target сүлжээг шинэчлэх
- Судалгаа, шинжилгээ хийх (Exploration)

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

Зураг 1.2: DDPG алгоритмын pseudo-code

Replay buffer

DDPG алгоритм нь replay buffer-ыг туршлагыг цуглуулахад ашигладаг. Цуглуулсан туршлагаа неороны сүлжээний параметрүүдийг шинэчлэхэд ашигладаг. Value болон policy сүлжээг шинэчлэхдээ replay buffer дахь туршлагуудаас санамсаргүй байдлаар цуглуулан ашигладаг.

Яагаад replay buffer-ыг ашиглаж байгаа вэ гэхээр алгоритд хамааралгүй байдлаар тархсан өгөгдөл хэрэгтэй. Ийм өгөгдлүүдийг replay buffer дахь туршлагуудаас санамсаргүй байдлаар сонгон авах байдлаар цуглуулж болно.

Actor (Policy) болон Critic (Value) сүлжээг шинэчлэх

Value сүлжээг шинэчлэх үйл явц нь Q-learning-тэй төстэй байдлаар хийгддэг. Шинэчлэгдсэн Q value-г Bellman-ны тэгшитгэлээс гарган авна:

$$y_i = r_i + \gamma Q^{'}(s_i + 1, \mu^{'}(s_i + 1|\theta^{\mu^{'}})|\theta^{Q'})$$

DDPG-д дараагийн төлөв Q утгуудыг target value network, target policy network ашиглан тооцдог. Дараа нь шинэчлэгдсэн Q утга ба анхны Q утга хоорондын дундаж квадрат алдааг хамгийн бага хэмжээнд хүртэл бууруулна:

$$Loss = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

Анхны Q value нь target network-ооc биш value network-ооc бодогдон гарна.

Policy функцын хувьд гол зорилго нь буцан ирэх үр дүн хамгийн дээд хэмжээнд байх юм:

$$J(\theta) = E[Q(s, a)|_s = s_t, a_t = \mu(s_t)]$$

Policy алдагдлыг тооцоолохын тулд зорилгын функцын деривативыг авна. Actor(policy) функц нь ялгагдах боломжтой тул гинжин дүрмийг хэрэгжүүлэх ёстой:

$$\bigtriangledown_\theta \mu J(\theta) \approx \bigtriangledown_a Q(s, a) \bigtriangledown_\theta \mu \mu(s|\theta^\mu)$$

Policy-гоо off-policy байдлаар шинэчлэж байгаа учир санамсаргүй байдлаар авсан туршлагуудынхаа градиентүүдийн нийлбэрийн дундаж утгыг авна:

$$\bigtriangledown_\theta \mu J(\theta) \approx \frac{1}{N} \sum_i [\bigtriangledown_a Q(s, a|\theta^Q)|_s = s_i, a = \mu(s_i) \bigtriangledown_\theta \mu \mu(s|\theta^\mu)|_s = s_i]$$

Target сүлжээг шинэчлэх

Target сүлжээний параметрүүдийг хуулбарлаад, тэдгээрээр дамжуулан сурсан сүлжээнүүдээ хянана:

$$\theta^{Q'} \longleftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$$

$$\theta^{\mu'} \longleftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$$

τ бол ихэвчлэн 1-тэй ойролцоо байхаар сонгосон параметр юм (жишээлбэл: 0.999).

Судалгаа шинжилгээ

Reinforcement learning буюу бататгасан сургалтад тасралттай үйлдлийн хувьд суралцах үйл явц нь санамсаргүй үйлдлийг сонгох замаар явагддаг. Харин үргэлжилсэн үйлдлийн хувьд суралцах үйл явц нь үйлдэлд шуугианыг нэмэх замаар явагддаг. DDPG алгоритмын баримт бичиг зохиогчид үйлдлийн үр дүнд дуу чимээ нэмэхийн тулд N:Ornstein-Uhlenbeck Process-г ашигласан байна:

$$\mu^{'}(s_t) = \mu(s_t|\theta_t^{\mu}) + N$$

Ornstein-Uhlenbeck процесс нь өмнөх дуу чимээтэй уялдаатай холбоотой шуугианыг бий болгодог.

1.2 Ашигласан технологи

Gym, pytorch etc

1.2.1 Gym

Gym бол reinforcement learning буюу бататгасан сургалтын алгоритмуудыг хөгжүүлэх болон харьцуулахад зориулагдсан хэрэгсэл юм. Үүнийг ашиглан агентдаа алхах, тоглоом тоглох зэрэг бүх зүйлийг зааж болно.

Яагаад үүнийг ашигладаг вэ?

Бататгасан сургалт (RL) нь шийдвэр гаргахтай холбоотой машин сургалтын дэд талбар юм. Энэ нь агент нарийн төвөгтэй, тодорхойгүй орчинд хэрхэн зорилгодоо хүрч болохыг судалдаг. RL нь доорх 2 шалтгааны улмаас ихээр ашиглагдаж байна:

- RL нь дараалсан шийдвэр гаргахтай холбоотой бүхий л асуудлыг багтаасан байдаг. Жишээ нь роботын хөдөлгүүрийг удирдаж, түүнийг үсрэх чадвартай болгох, үнэ, бараа

материалын менежмент гэх мэт бизнесийн шийдвэр гаргах, видео тоглоом, ширээний тоглоом тоглох гэх мэт

- RL алгоритмууд олон хүнд хэцүү орчинд сайн үр дүнд хүрч эхэлсэн

Гэсэн хэдий ч RL судалгааны ажлыг хоёр хүчин зүйл удаашруулж байна:

- RL-ын open-source орчин хангалттай олон янз байдаггүй бөгөөд тэдгээрийг тохируулах, ашиглахад хэцүү байдаг.
- Орчны стандартчилал дутмаг.

Gym нь эдгээр 2 асуудлыг шийдэхийг зоридог.

2. ХЭРЭГЖҮҮЛЭЛТ

3. ҮР ДҮНГИЙН БОЛОВСРУУЛАЛТ

3.1 Туршилтын үр дүн

3.2 Үр дүнгийн харьцуулалт

Дүгнэлт

Дүгнэлтийг энд бич

Bibliography

[1] Deep Deterministic Policy Gradients Explained, TowardsDataScience, https://towardsdatascience.com/deep-deterministic-policy-gradients-explained-2d94655a9b7b

[2] Deep Deterministic Policy Gradient (DDPG), Keras, https://keras.io/examples/rl/ddpg_pendulum/

[3] Deep Deterministic Policy Gradient, Spinning Up, https://spinningup.openai.com/en/latest/algorithms/ddpg.html

[4] Continuous Control With Deep Reinforcement Learning, Lillicrap et al 2015, https://arxiv.org/pdf/1509.02971.pdf

[5] Tables, Share LaTex, https://www.sharelatex.com/learn/Tables

A. A

Хавсралтын агуулга

В. КОДЫН ХЭРЭГЖҮҮЛЭЛТ