Reinforcement Learning ашиглан автомат удирдлагатай тэргэнцрийн алгоритмыг хэрэгжүүлэх

Гантулга Гантөмөр
Электроникийн салбар
Шинжлэх Ухаан Технологийн
Сургуулийн Мэдээлэл Холбооны
Технологийн Сургууль
Улаанбаатар, Монгол Улс
limited.tulgaa@gmail.com

Луубаатар Бадарч
Электроникийн салбар
Шинжлэх Ухаан Технологийн
Сургуулийн Мэдээлэл Холбооны
Технологийн Сургууль
Улаанбаатар, Монгол Улс
luubaatar@must.edu.mn

***Хураангуй* — Сүүлийн жилүүдэд хиймэл оюун ашиглан хүн төрөлхтний өмнө тулгарч буй олон асуудлуудыг шийдвэрлэж байна. Үүнд мөн хөгжлийн бэрхшээлтэй иргэдэд тулгамдаад буй асуудлуудыг шийдвэрлэхийг оролдож байгаа юм[5]. Машин сургалтын аргуудын нэг салбар болох *Reinforcement Learning (RL)* аргын судалгаа болон хэрэгжүүлэлт өргөн хийгдэж байна. Бид энэхүү ажлаар бусдын тусламжтайгаар тэргэнцэр ашиглан явдаг хүнд зориулж *RL* ашиглан тодорхой хийсвэр орчинд өөрөө явдаг тэргэнцрийн алгоритмыг хэрэгжүүлж үзсэн болно. RL ашиглан сургалт явуулсны үр дүнд агент хэдий дөт замаар явж байсан ч үүссэн зам нь хэт замбараагүй байх тохиолдол олон байсан. Үүнийг бид тодорхой алгоритм хэрэгжүүлэн засаж чадсан. Мөн сургалтын явц сургаж буй орчноос хамааран хэт удах, зорилгодоо хүрч чадахгүй байх зэрэг асуудал гарсныг бид сургалтын параметрүүдийг өөрчлөх замаар шийдэж чадсан.**

Түлхүүр үгс **— Reinforcement Learning, Q-Learning, Learning Rate, Discount Rate, Reward, Agent, State, Action Environment, Q-Table, Hyperparameter**

I. Оршил

Reinforcement Learning нь машин сургалтын үндсэн 3 аргуудын нэг юм. Энэхүү сургалтын арга нь сүүлийн жилүүдэд технологийн хурдтай хөгжил компьютер ийн тооцоолох чадвар асар их нэмэгдсэн зэргээс шалтгаалан хиймэл оюуны салбарт өндөр үр дүн үзүүлэх болсон. *RL* нь бэлэн өгөгдөл дээр бус өгөгдлийг агуулж буй орчинд хандаж ажилласнаар суралцдаг ба энэ нь бидний хэрхэн хүрээлэн буй орчинтойгоо харьцаж шинэ зүйлсийг суралцдагтай төсөөтэй хэрэгждэг арга юм[1]. *RL*–ыг Robotics, Өөртөө удирдлагатай машин, Видео тоглоом зэрэгт өргөн ашиглаж байгаа бөгөөд бидний өдөр тутамдаа хийдэг үйлдлүүдийг хөнгөвчлөх, автоматжуулах гэх мэт зорилгоор маш олон судалгааны ажлууд хийгдэж байна. *RL*–ыг ашиглан сургасан томоохон жишээнүүд нь *DeepMind–AlpaGo, StartCraft-II*[2], *OpenAI–Dota2 OpenAI Five*[4] зэрэг юм. *RL*-ийн агент нь өөрийн хүрээлэн буй орчинд тодорхой нэг үйлдэл(action) хийх ба үр дүнд агентын нөхцөл байдлаас хамааран тодорхой хэмжээний шагнал (*reward-r*) авна. *r* ямар байхаас хамааран тухайн нөхцөл байдалд(*state*) ямар үйлдэл(*action*) хийгээд ямар нөхцөл байдалд(*state'*) орсноос хамааран дүгнэх замаар агентыг сургадаг. Ийм бүтцээр задалж үзсэнийг MDP(*Markov Decision Problem*)[3] гэж авч үздэг. Бидний хийсэн орчин нь тэргэнцэр ашиглаж буй хүний амьдардаг орчин гэж төсөөлж хийсэн бөгөөд тухайн орчныг үнэн зөв дүрсэлж чадвал орчин ямар байхаас үл хамааран суралцах боломжтой. Бид *LR*-ын *Q-Learning* аргыг ашиглан сургалт явуулсан.

A. Environment (Сургалтын орчин)

Сургалтын орчин нь агент байрлах хэсэг бөгөөд агентын зорилго $\{S_T\}$, нөхцөл байдал $\{S\}$, хийж болох үйлдлүүд $\{A\}$ болон шагнал $\{R\}$ зэргийг тодорхойлсон байна[1][3.1].

B. Agent (Агент)

Агент нь тодорхой зорилгод хүрэхийн тулд тухайн орчноос хамааран дараагийн хийх үйлдлээ тодорхойлон гүйцэтгэх үүрэгтэй[1][3.1].

C. State (Нөхцөл байдал)

Агентын суралцах орчноос авсан мэдээлэл ба агентын зорилгоос хамааруулж бид ямар мэдээлэл авч болохыг тодорхойлно. OpenAI–ын гаргасан Dota2 тоглоомыг хүний түвшинд тоглодог Dota2 Five агентуудыг тоглоомын пикселийг агентын нөхцөл байдал болгон сургасан байдаг[4].

D. Action (Үйлдэл)

Агентын суралцах орчноос авсан мэдээлэлд үндэслэн хийж буй үйлдэл. Үйлдлийг агентын тухайн нөхцөл байдалд хийж болох үйлдлүүдээс сонгон хийнэ. Жишээлбэл DeepMind–ийн StarCraft-II[2]–ын агентын хийж болох үйлдлүүд нь тоглоомын дүрүүдийг хөдөлгөх, сонгох ,барилга барих, дайрах гэх мэт байгаа юм.

E. Reward (шагнал)

Шагнал нь агентын хийж буй үйлдлийг үнэлэх зорилготой ба агент муу үйлдэл хийвэл харьцангуй бага, сайн үйлдэл хийвэл харьцангуй их шагнал авна. Үнэлэх утгуудыг агентын зорилгоос хамааруулан олгоно[1][3.2]. OpenAI–Dota2 Five сургалтын орчны шагнал нь тоглогчдыг ганцаар бус багаар ажиллах үйлдлийг илүүд үзэх зорилготой байх гэх мэт[4][G].

II. Q-Learning

Q–Learning нь "*model free RL*" алгоритм юм. *Q-Learning*–ийг MDP гэж тооцогддог бүх асуудлуудад ашиглаж болох юм. Энэ нь "*model free RL*" алгоритмын давуу тал юм. Агентын байж болох бүх нөхцөл байдлуудад хийж болох үйлдлүүдийг үнэлсэн хүснэгт (*Q-Table*) байх ба *Q–Table* нь агентын хийх үйлдлийг заадаг тархи гэж ойлгож болох бөгөөд *Q–Table*–ийг агентын зорилгод тааруулж өөрчлөхөд *Q-Learning*–ийн гол утга учир оршино. Сургалтыг хэрэгжүүлэхийн тулд "*Bellman Equation*"–ийг ашиглана[1][6.5].

$$Q_{(s,a)} = r_{(s,a)} + \gamma * \max_{a'} Q_{(s',a')} \qquad (1)$$

Үүнд:

s' – үйлдэл хийсний дараах агентын нөхцөл байдал.

a' – үйлдэл хийсний дараах агентын нөхцөл байдалд хийж болох үйлдлүүд.

γ – *discount rate*.

$r_{(s,a)}$ –өмнөх нөхцөл байдлаас үйлдэл хийсний дараа авсан шагнал.

Bellman Equation нь агентын тухайн нөхцөл байдлаас үндэслэн хамгийн зөв гэж дүгнэж буй үйлдлийг хийсний дараа агентын шинэчлэгдсэн нөхцөл байдалд хийж болох үйлдлүүдийг дүгнэсэн үнэлгээний хамгийн ихийг авч тухайн хийсэн үйлдлээ дүгнэнэ.

Агентын зорилго нь өөр өөр байж болох ч *Q-Learning* нь агентыг хугацааны төгсгөлд хамгийн их шагнал авдаг байхаар сургана[1][3.1].

$$R = r_t + \gamma * r_{t+1} + \gamma^2 * r_{t+2} + \cdots + \gamma^n * r_{t+n} \quad (2)$$

$$R \rightarrow R_{max}$$

R – Нэг үеийн турш цуглуулсан шагнал.

Шагналыг хамгийн их байлгана гэдэг санаа нь агентыг зорилгоос нь хазайлгах магадлалтай бөгөөд үүнээс зайлсхийхийн тулд сургалтын параметрууд буюу "*Hyperparameter*" –уудыг зөв тааруулах хэрэгтэй[1][1.4].

Q-Learning нь *Off Policy* learning алгоритм юм. Энэ нь агентын сургалтын орчинд хийх үйлдэл нь ямар нэг гаднын хүчин зүйлээс хамаарахгүй гэсэн үг юм[1][5.5].

III. HYPERPARAMETERS

Machine Learning –ийг хэрэгжүүлэхэд чухал шаардлагатай хүчин зүйл болох *hyperparameter* нь суралцах үйл явцыг тодорхойлдог мэдээлэл юм. [1]-д hyperparameter-уудын утга сургалтын явцад нөлөөлөх нөлөөллүүдийг харуулахад γ тухайн сургалтын орчинд 1 тэй тэнцүү байх нь буруу сонголт гэдгийг харуулж байна. Хүмүүс бид *Hyperparameter* –уудыг сургалтанд аль болох тааруулж тодорхойлдог.

Reinforcement Learning –д:

- *Reward (reward - r)*
- *Discount rate (gamma - γ)*
- *Learning rate (alpha - α)*
- *Initial Q-Table*
- *Randomization (epsilon - ε)*

зэрэг нь *hyperparameter* –ууд юм.

A. Discount rate - γ

γ нь агентын цуглуулах шагналыг тодорхой хязгаарт барих үүрэгтэй. Өмнө хийсэн үйлдлийг тухайн нөхцөл байдалд хийж болох үйлдлүүдийн үнэлэлтээс хэр их хамаарахыг зааж өгнө.

γ их байх нь харьцангуй холын үйлдлүүдээс хамаарч үнэлгээг хийнэ гэсэн санаа юм. Сургалтын орчин, шагнал, агентын зорилгоос хамаардаг.

B. Learning rate - α

α нь *Q-Table* –ийг ирээдүйд авч чадах шагналаас хэр их хамааруулж өөрчлөхийг зааж өгнө[1][1].

C. Q-Table

Сургалтын орчин дахь бүх нөхцөл байдалд хийж болох үйлдлүүдийг үнэлэх мэдээллийг агуулсан функц юм[3][7].Сургалтыг эхлэхэд *Q-Table* –ийг тодорхой утгуудтайгаар эхэлж болох бөгөөд тэдгээр нь суралцах явцад том нөлөө үзүүлнэ.

D. Randomization

Reinforcement Learning –ийн бас нэг том хүчин зүйл нь Exploration юм. Энэ нь агентыг аливаа нөхцөл байдалд үнэлгээнээс хамаарахаас илүү хийж үзээгүй зүйлээ хийхийг санал болгох буюу хүнээр сониуч хүнтэй зүйрлэж болох юм. Exploration & Exploitation –ыг суралцах орчинд тааруулж тогтвортой байлгаж чадвал илүү үр дүнг үзүүлэх магадлалтай гэж үздэг[1][1.1] ба зарим тохиолдолд заавал байх ёстой зүйлийн нэг байдаг ч зарим тохиолдолд байхаасаа байхгүй байсан нь илүү үр дүн үзүүлж болох юм.

IV. RL GRID WORLD

Бидний бүтээсэн *Grid World* нь нэг цэгээс саадуудыг тойрон тодорхой нэг цэгт хамгийн дөт замаар очих даалгавартай бөгөөд агент нь суралцах орчны тухай ямар ч мэдээлэлгүйгээр (*Q-Learning*) суралцах ба хийсэн үйлдэл бүрдээ харгалзах шагналуудыг авах юм. Шар өнгөөр агент, ногоон өнгөөр байг, улаан өнгөөр саадыг дүрсэлсэн болно. Энэхүү орчны бүтцийг өөрчлөн даалгаврыг хүндрүүлэх замаар *Hyperparameter* –уудын хамаарлыг гаргаж авсан. Энэхүү хамаарлуудыг зөв гаргаж авснаар сургалтын үр дүн, суралцах хугацааг хэмнэх боломжтой. Ингэснээр тэргэнцэр ашиглаж буй хүний цагийг хэмнэх, тэргэнцэр нь хурдан шуурхай ажиллах боломжтой болох юм. Сургалтын орчныг тэргэнцрийг ашиглаж эхлэхээс өмнө тэргэнцрийн удирдах хэсэгт тодорхойлж өгөх хэрэгтэй.

Тоглоомын төлвийг параметрүүдтай уялдуулахын тулд эхний суралцах орчны төлвийг тооцоолоод дараа дараагийн төлвүүдийг нэг нэгээр нэмэгдүүлэн тухайн төлөв бүрд харгалзах суралцах орчнуудаг гаргаж авсан. Ингэхдээ:

$b_{size} = a * a$ - Тоглоомын талбарын хэмжээ
(a – талын урт)

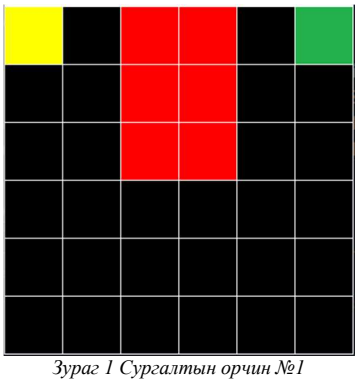
obs - Талбар дээр байх саадын тоо

act - Тоглоомын state бүрд хийж болох үйлдлийн тоо

$\frac{b_{size}}{obs} = 6$ байхаар *obs*-ийг сонгосон.

$$C = \ln(b_{size} * obs + b_{size} * act + obs * act) \quad (3)$$

C – Complexity score



Зураг 1 Сургалтын орчин №1

Дараа дараагийн сургалтын орчнуудыг олсон нь

$$a^4 + 28a^2 - 6C = 0 \qquad (4)$$

Хүснэгт 1. Сургалтын орчин

№	*C*	*a*	*obs*	*act*
1	6	6	6	4
2	7	8	11	4
3	8	11	20	4
4	9	14	33	4
5	10	19	60	4
6	11	24	96	4
7	12	31	160	4
8	13	40	267	4

V. СУРГАЛТЫН ҮЙЛ ЯВЦ

Сургалтын орчин - №1

Зураг 1 дээрх сургалтын орчин нь хамгийн бага төлвийн оноотой буюу $\boldsymbol{C} = 6$ оноотой.

Анхны параметрүүдийн утга:

Reward function:

$$R = \begin{cases} if\ agent == target \rightarrow r = 2 \\ if\ agent == hole \ \rightarrow r = -2 \\ \quad else\ (-1)/((a-1)*4) \end{cases} \qquad (5)$$

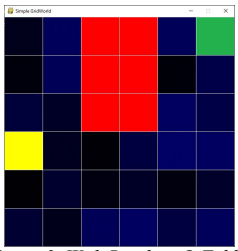
γ = 0.9 Бидний Gridworld орчинд агент алсыг харж тухай бүрийн үйлдлээ хийх учир γ –ыг их байхаар авсан болно[1-Returns].

α = *None* (Хамгийн зөв утгыг олох)

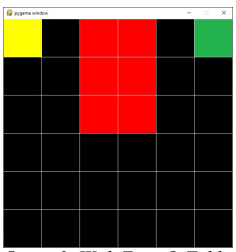
Initial Q-Table = np.zeros(a, a, act) (бүх q-value –ууд 0) *Initial Q-Table*–ийг 0 ээр авч рандомчлалаас татгалзсан нь хэд хэдэн удаагийн оролдлогоор сургалтын зорилго үр дүнд сөргөөр нөлөөлж байгаа нь харагдсан.

ε = *0*

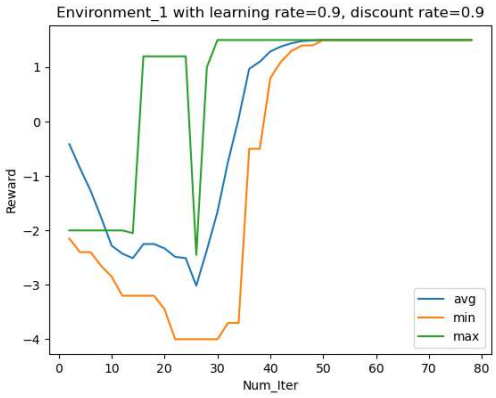
Агентыг тогтвортой байлгаж бидний хийж буй туршилтыг үр дүнтэй болгохын тулд тоглоомоос рандомчлалыг хасаж тооцож байгаа.



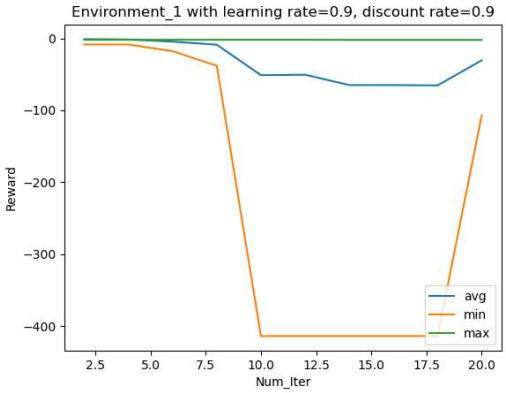
Зураг 2. With Random Q-Table



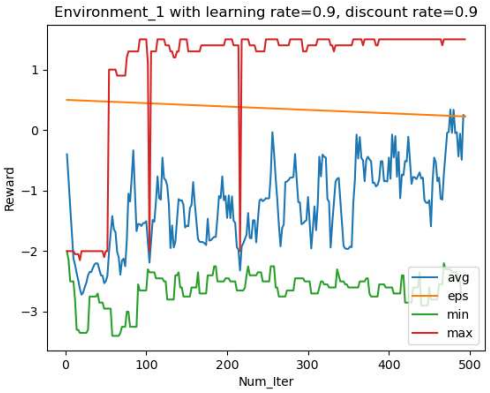
Зураг 3. With Zero Q-Table



Зураг 4. Without Random Q-Table and ε



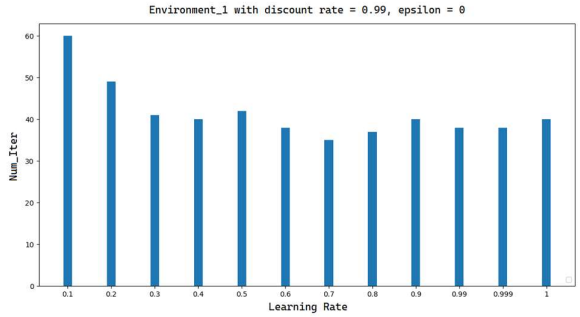
Зураг 5. With Random Q-Table



Зураг 6. With Random Q-Table and ε

Дээрх гурван графикаас харахад дан *Random Q-Table* –ийг агентыг сургахад ашиглаж болохгүйг харуулж байна. *Random Q-Table + Epsilon Greedy* аргуудыг хослуулж үзэхэд агент сургалтын оролдлогын эхэн хэсэгт сурсан ч ε байгаа учир тогтворгүй явсаар олон оролдлогын дараа тогтворжиж байна. Энэ нь ε нь сургалтын хугацаанд сөргөөр нөлөөлөхийг шууд харуулж байна. Зөвхөн тохирсон α болон γ зэргийг ашиглан хурдан хугацаанд сургаж болохыг харуулж байна.

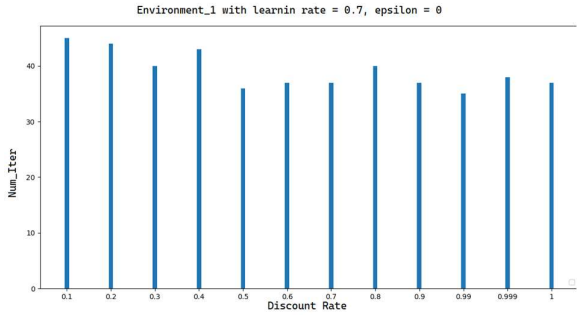
Grid Search – ашиглан хамгийн зөв α –ыг олох нь.



Зураг 7. α range

Өөр өөр α аар нийт 12 удаа сургасны үр дүнд $\alpha = 0.7$ байхад хамгийн бага үе тоглосон байна.

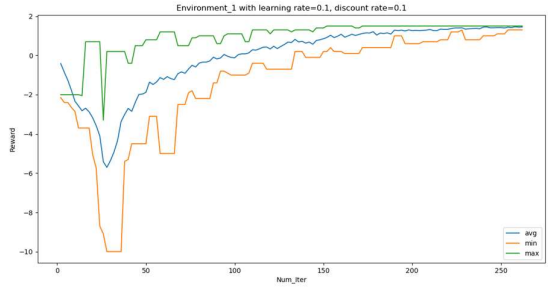
Grid Search – ашиглан хамгийн зөв γ –ыг олох нь.



Зураг 8. γ range

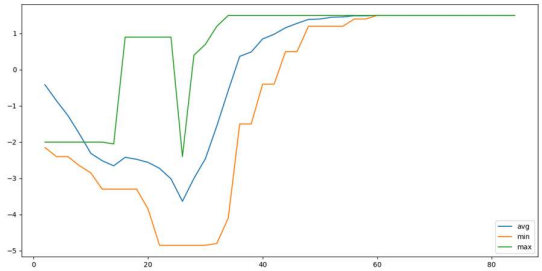
Мөн өөр өөр γ аар нийт 12 удаа сургаснаас үр дүнд $\gamma = 0.99$ үед хамгийн бага үе тоглосон байна.

Дээрх мэдээллийг ашиглан агентыг тохирох параметртэй үед болон үл тохирох параметртэй үед ямар ялгаа гарахыг харвал тохирох параметртэй үед агент хэвийн сурч байгаа бол үл тохирох параметтэй үед агент хэт удаан сурч байгааг харж болно.



Зураг 9. Тохироогүй параметртэй

Сургалт удаан явж байгаа шалтгааныг дээрх графикаас харж болно. Эндээс α хэт бага байснаар агентын гаргасан алдааг маш бага хувиар авч тухайн үйлдлээ дүгнэснээр тогтворжилт удаан явагдаж байна.



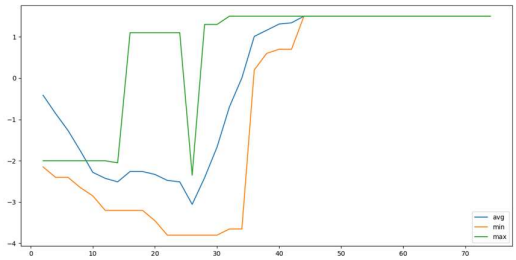
10. Тохироогүй γ тохирсон α

$$L = \mathbb{E}[r_{(s,a)} + \gamma * \max_{a'} Q_{(s',a')}] - Q_{(s,a)} \quad (6)$$

Дээрх тэгшитгэл агентын гаргасан алдааг бодно.

$$Q_{(s,a)} = Q_{(s,a)} + \alpha * L \quad (7)$$

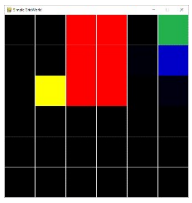
Өмнө тооцоолсон алдааг ашиглан *Q-Table* –ээ шинэчлэхдээ алдааг $(1 / \alpha)$ дахин багасгаж байгаа нь хэрэв α бага байвал агентын хийсэн алдааг бууруулна. Ингэснээр жинхэнэ *Q-Table* –ын утга руу удаан дөхнө.



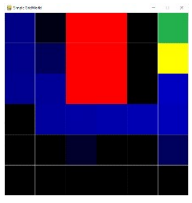
Зураг 11. Тохирсон параметртэй

Мөн дээрх графикт γ–ийн буруу сонголт ямар нөлөөтэйг харж болно. Агент нь бай руу олон төрлийн замаар очиж байна. Энэ нь бидний үүсгэсэн сургалтын орчинд γ *нь* агентын байн дээр очих замыг бусад замаас ялгаж өгдөг гэж ойлгож болох юм. Бага байвал агент сургалтын орчинд удаан хугацаанд хайлт хийсний дараа байнд хүрэх дөт замыг олох магадлалтай болж байгаа юм.

Доорх зурагт бага болон их γ –ийг сурсан агент дээр харууллаа.



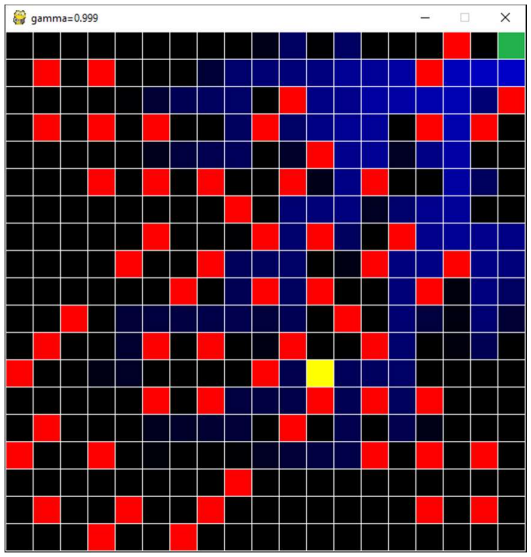
12. Бага γ



13. Их γ

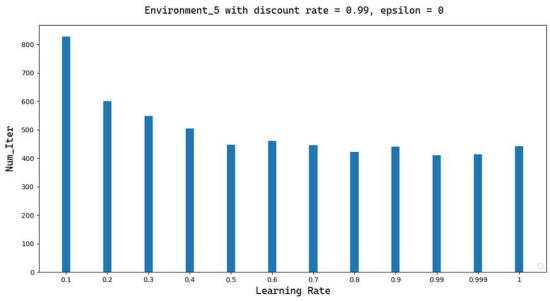
Сургалтын орчин - №5

C = 10 Энэ талбар өмнөх талбараас харьцангуй хэцүү ба γ –ын хэмжээ их байх ёстой гэдэг нь суралцсан үр дүнгээс харагдаж байна.



Зураг 14. Суралцах орчин 5 сурсан агент

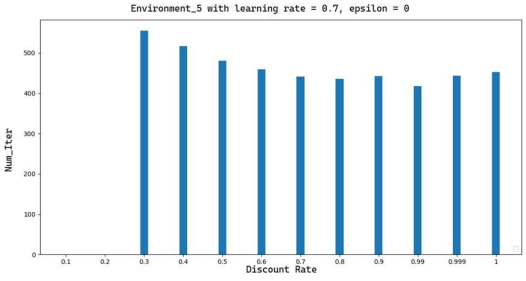
Grid Search – ашиглан хамгийн зөв α –ыг олох нь.



Зураг 11. α *range*

Хамгийн бага давталт хийсэн – 0.99
Хамгийн их давталт хийсэн – 0.1

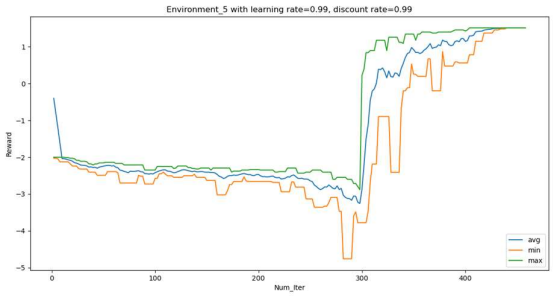
Grid Search – ашиглан хамгийн зөв γ –ыг олох нь.
Хамгийн бага давталт хийсэн – 0.99



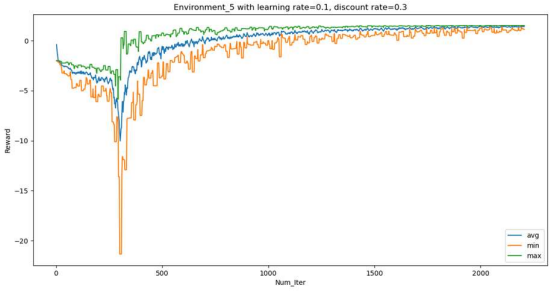
Зураг 12. γ *range*

Хамгийн их давталт хийсэн – 0.3 (0.1; 0,2 – failed at training)

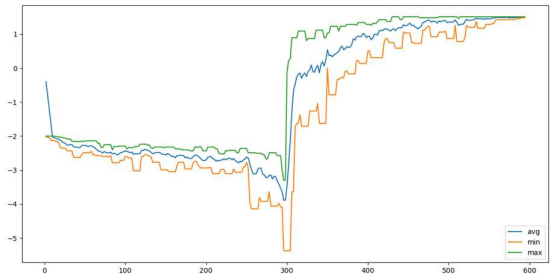
Сургалтын орчин 5 –аас эхлэн параметрүүдийн доод хязгаар харагдаж эхэлж байна.



Зураг 13. Тохирсон параметртэй



Зураг 14. Тохироогүй параметртэй



Зураг 15. Тохироогүй γ *тохирсон* α

Энэхүү жишээн дээр агент 300 дахь оролдлого орчимд бай руу хүрч байгаа боловч γ –ын ялгаанаас болж тогтворжих хугацаа харилцан адилгүй байна. Үүнийг хүнээр төсөөлбөл γ их бол агент өөртөө итгэлтэй болох бөгөөд энэ нь агентыг олон төрлийн зам биш цөөхөн зам дагаж явах боломжийг олгож байна. Өөрөөр хэлбэл агент буруу үйлдэл хийсэн бол тухайн үйлдлийг маш буруу үйлдэл хийлээ гэж дүгнэх боломжтой болж байгаа юм.

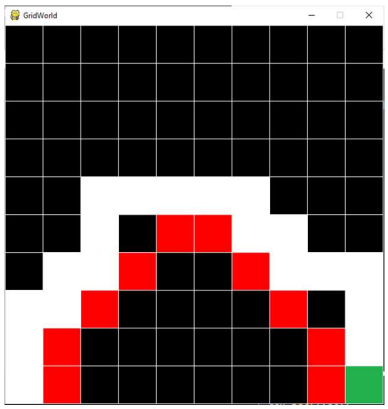
Дээрх аргуудыг ашиглан нийт 8 сургалтын орчинд хэрэгжүүлж үзсэн. Үр дүнд

Хүснэгт 2. Best and Worst Values

№	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
C	6	**7**	8	9	10	11	12	13
Alpha-optimal	0.7	0.7	0.7	0.7	0.99	0.999	0.999	1
Gamma-optimal	0.99	0.3	0.9	0.8	0.99	0.999	0.999	0.8
Alpha-bad	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Gamma-bad	0.1	0.99	0.1	0.1	0.3	0.2	0.4	0.5
Best Actions	45	72	170	155	450	520	730	420

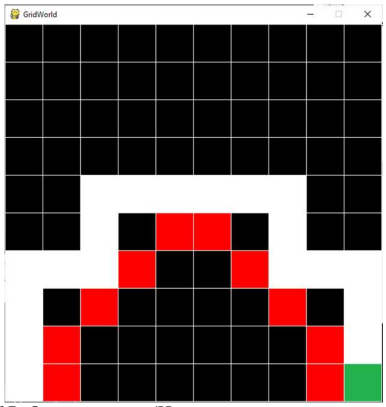
VI. Оновчой зам сонгох

RL ашиглан тухайн орчинд гарган авсан зам нь хамгийн дөт зам ч тэргэнцэр явахад оновчтой зам биш болохыг доорх зурагнаас харж болно.



Зураг 16. Оновчгүй зам(Цагаанаар аггентын явсан замыг харуулав)

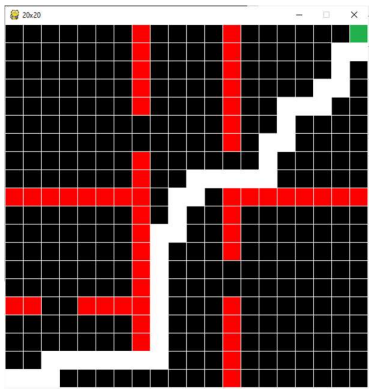
Дээрх замыг явж буй хүнд хүндрэлгүй мөн дөт байлгах шаардлагатай. Үүнийг бид хялбар алгоритм ашиглан шийдэж чавсан.



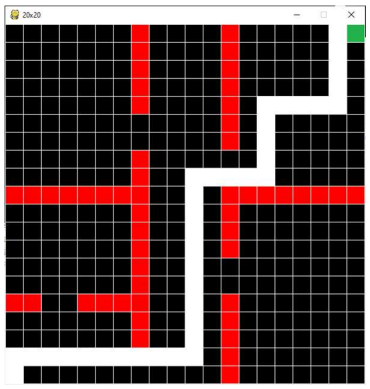
Зураг 17. Оновчтой зам(Цагаанаар аггентын явсан замыг харуулав)

VII. Дүгнэлт

RL-ыг *Gridworld* орчинд хэрэгжүүлж хөгжлийн бэрхшээлтэй хүний тэргэнцрийг автоматжуулж болохыг амжилттай харууллаа. Сургалтын явцыг хурдасгах агентын чиглэл замыг засах зэрэг ажлуудыг хийж гүйцэтгэлээ. Цаашдаа энэхүү алгоритмыг бодит орчинд хэрэгжүүлэхийн тулд нарийн техник хангамжийн судалгаа хийх шаардлагатай ба тэргэнцрийг амар хялбар ашиглах зорилгоор техник, програм хангамжийн асуудлуудыг шийдэх шаардлагатай. Жишээлбэл хэрэглэгчид хялбар байлгах үүднээс мэдрэгч бүхий дэлгэц, ойлгоход хялбар үйлдлийн систем интерфейс зэргийг шийдэж өгөх хэрэгтэй. Мөн тэргэнцрийн урд таарах таамаглаагүй саадуудыг тойрж хэрэгтэй ба үүнийг мөн машин сургалт ашиглан шийдэж болох юм.



Зураг 18. Оновчгүй зам(Цагаанаар аггентын явсан замыг харуулав)



Зураг 19. Оновчтой зам(Цагаанаар аггентын явсан замыг харуулав)

VIII. Ашигласан материал

[1] Richard S. Sutton and Andrew G. Barto “Reinforcement Learning: An Introduction” Second edition, in progress in The MIT Press Cambridge, Massachusetts London, England.

[2] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Kuttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap - DeepMind Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, Rodney Tsing – Blizzard “StarCraft II: A New Challenge for Reinforcement Learning”.

[3] Abhijit Gosavi “A Tutorial for Reinforcement Learning” - 1 Introduction in Department of Engineering Management and Systems Engineering Missouri University of Science and Technology 210 Engineering Management, Rolla, MO 65409.

[4] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław “Psyho" Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, Susan Zhang “Dota 2 with Large Scale Deep Reinforcement Learning” December 13, 2019

[5] Tamei, Tomoya, et al. "Reinforcement learning of clothing assistance with a dual-arm robot." 2011 11th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2011.