СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ”

СТОПАНСКИ ФАКУЛТЕТ

Есе по дисциплината

Основи на изкуствения интелект:
Машинно самообучение с подкрепа (Reinforcement learning)

Изготвил:

Слав Йолов
№ 5EB3100235

Преподавател:

доц. д-р Ангел Ангелов Марчев

София
Януари 2023

Съдържание

Въведение

Една от основните цели на изкуствения интелект е да създаде напълно автономни агенти, които взаимодействат със заобикалящата ги среда, за да научат оптималното поведение посредством множество итерации от тип проба-грешка. Създаването на такъв тип решения е дългогодишно предизвикателство, което се простира от хардуерни решения (робот, който извършва поредица от различни задачи) до чисто софтуерни такива (алгоритми, които са в състояние да победят световни шампиони при игра на шах, го, дота 2 и други). Общото между тях е, че могат да се групират принципно под названието Машинно самообучение с подкрепа (Reinforcement learning), което е итеративен автономен процес за учене посредством трупане на опит.

Целта на настоящето научно есе е запознаване с основните термини при Reinforcement learning, както и да разгледа един последователен процес за вземане на решения използвайки Марковски процеси.

Дефиниция и основна терминология

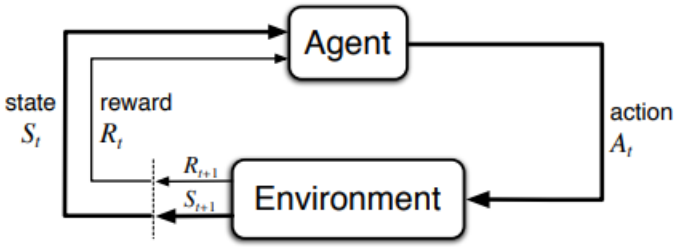
Sutton и Barto (2018, pp. 1-2), определят reinforcement learning (RL) като процес на учене, при който имаме за задача да максимизираме цифров награден сигнал посредством вземане на действия. Ученикът (още наричан агент) не знае кои действия са правилни, но посредством множество опити трябва да разбере тези, които дават най-голяма награда. В определени казуси, предприетите действията не само определят моментната награда, но също така и следващата ситуация и всички последващи награди. От това могат да бъдат изведени две основни характеристики при (RL)

- Постигането на оптимизация става посредством процес от тип проба-грешка
- Получаването на кумулативната награда е забавено във времето

Част от основните термини при RL са (Anwar, 2021):

- Агент (Agent) : обект, който наблюдава и изследва дадена среда и предприема определени действия
- Среда (Environmen) : мястото от което агентът е обграден и където той съществува
- Действия (Actions) : ходовете, който агентът предприема в заобикалящата го среда
- Състояние на средата (Environment State) : състоянието е ситуацията върната от средата след всяко действие предприето от агента.
- Състояние на агента (Environment State) : информацията която агента използва, за да избере своето следващо действие
- Награда (Reward) : обратна връзка върната от средата към агента, която позволява оценка на действието на агента.
- Политика (Policy): стратегия прилагана от агента за следващото действие базирано на текущото състояние
- Полза (Value) : очакваната дългосрочна полза от действията на агента

Взаимовръзката между агента и средата и техните входове и изходи са показани на Фигура 1.



Фигура 1 : Sutton и Barto (2018, p. 48), The agent–environment interaction in a Markov decision process

Пример за RL при затворена напълно обозрима система е игра на шах. В нея агентът играе срещу опонент (човек или друг агент). Ходовете на агента са фигурките преместени по дъската, а като обратна връзка той получава награда и новото състояние на дъската. Наградата може да е позитивна, например когато спечели фигурка или при победа в играта. А също така може да е и негативна, когато загуби фигурка или самата игра. Възможна е различна точкова дефиниция на наградата, например +/- 100 точки за спечелена/загубена игра; +/- 20 точки при отнета царица; +/- 5 точки при офицер и т.н. Политиката на агента трябва да е смислена – при този пример победа в играта. Описаният пример е на практика последователен процес за вземане на решение.

Разглеждане или експлоатиране

Една от дилемите при изграждането на решение от тип машинно обучение с подкрепа е колко случайни да бъдат действията на агента. При разглеждането имаме възможността да открием нови потенциално по-добри възможности, докато при експлоатирането прилагаме най-доброто решение отново и отново. Когато се тренира невронна мрежа може да контролираме това чрез така наречения епсилон като той може да бъде статичен (epsilon greedy) или динамичен (decayed epsilon greedy). При статичния избираме процент шанс за разглеждане като той не се променя през процеса на трениране (например 30%) (Parkinson, 2019), докато при динамичния в началото оставяме модела да избира ходове на случаен принцип с висока вероятност (например 95%) като с всеки следващ епизод намаляваме стойността до достигане на дефинирана долна граница (например 10%) (Chinnamgari, no date).

Последователен процес за вземане на решения

Както Silver (2015a, p.18) посочва, агентът и средата взаимодействат посредством множество епизоди (цикличен процес). Тяхното взаимодействие формира история, която е поредица от наблюдения (O), действия (A) и награди (R)

$$H_t = O_1, R_1, A_1, \dots, A_t, O_t, R_t$$

Какво ще се случи в дадена стъпка зависи от историята H_t :

- Агентът избира действие
- Средата емитира наблюдение/награда

Освен историята, за да се определи какво следва ни е необходимо и състоянието S_t. Състоянието е информацията, която се използва, за да се определи какво следва като то е функция от историята $S_t = f(O_t)$. Дефиницията за функцията не е фиксирана като са възможни различни опции. Може да се използва цялата история, само предходното наблюдение или пък поредица от предходни наблюдения. Практиката показва, че всеки отделен проблем сам формулира тази функция, например при решението на Atari Breakout посредством Q-Learning са използвани последните четири фрейма (предходни наблюдения), за да се създаде агент, който може да максимизира получените точки (Mnih, V. et al., 2013). Избирайки този прагматичен подход, агентът трениран от Mnih, V. et al. (2013) е в състояние да разбере динамиката на средата - посоката на движение на топчето и скоростта му на движение. Изхождайки от това може да заключим, че настоящето информационно състояние (също познато като Марковско състояние) се състой от цялата полезна информация от историята Silver (2015a, p.21).

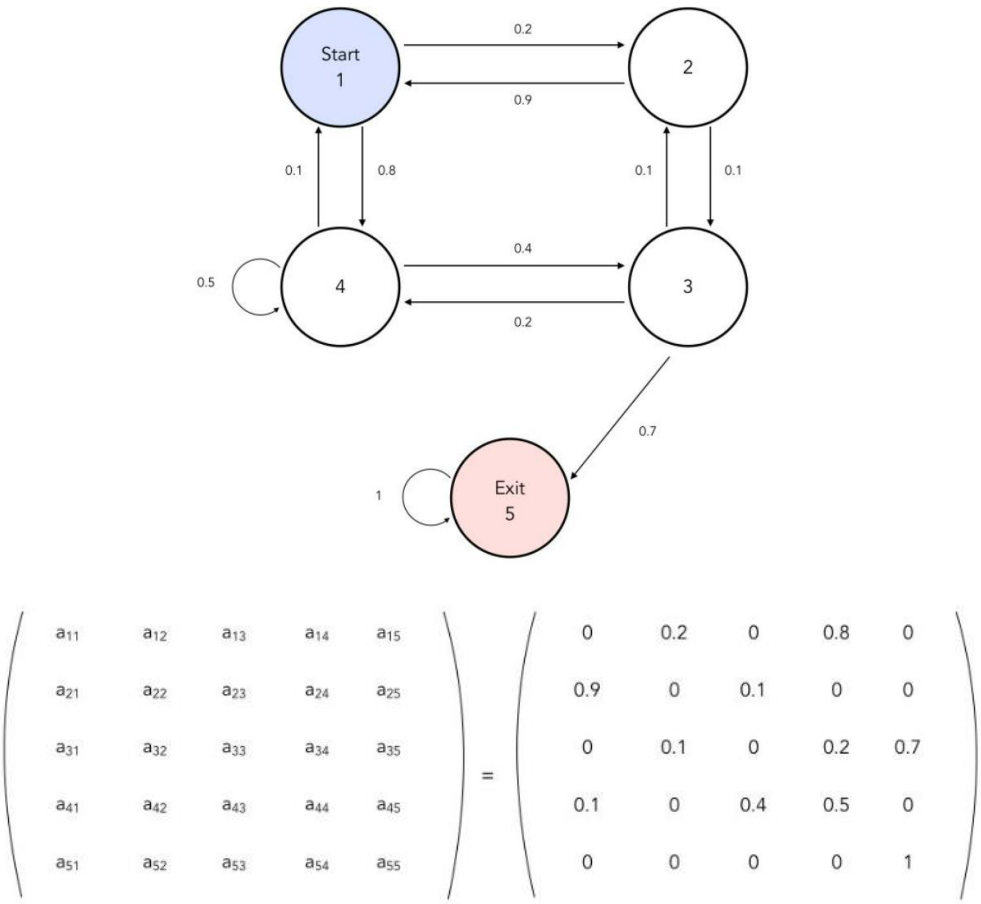
Вземане на решения посредством Марковски процеси

Голям принос за развитието на машинното самообучение с подкрепа има теорията дефинирана от Марков. Значителна част от проблемите решавани посредством Reinforcement learning се базират на т.нар. Марковски процес (Марковски вериги).

Марков, цитиран в (Madani, 2020), извежда следната дефиниция за бъдещето: Бъдещето е независимо от миналото имайки настоящето[1]. Това означава, че настоящето съдържа всичката важна информация за нашето решение. Например, това дали човек е гладен сега е ирелевантно с това дали е бил гладен преди два или пет дни. На базата на това, както Silver (2015a, p.21) посочва, можем да изведем, че знаейки текущото състояние вече нямаме нужда да пазим историята. Това от своя страна ни позволява да улесним процеса на учене и да го направим по ефективен, защото агентът не трябва да пази и анализира голямо количество информация, а трябва да се фокусира само върху текущото състояние. Това от своя страна го прави и по предсказуем в следващите действия, които би предприел (Madani, 2020).

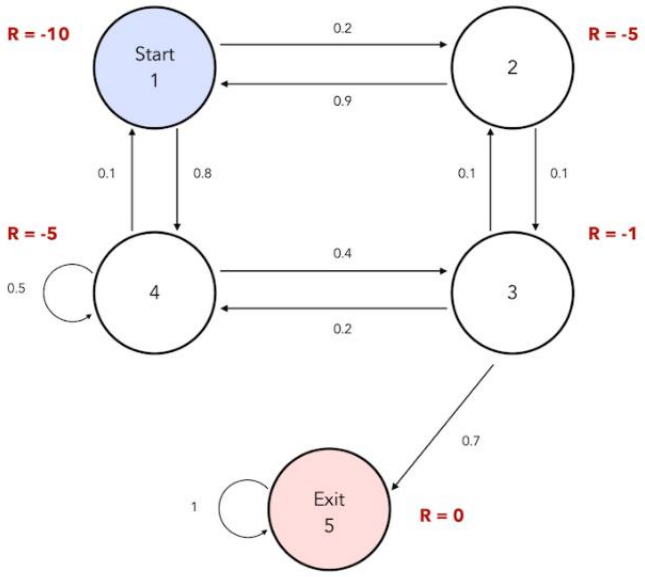
Веригите на Марков представлява набор от състояния с вероятност за преход от едно състояние в друго. Пример за такъв процес е, ако в даден момент t учиш за изпит и си завършил тема 2 от конспекта вероятността от това да продължиш да учиш тема 3 може да бъде дефинирана като 40%, а тази да провериш своя фейсбук 50%, а тази да се откажеш от учене да е 10% . Да допуснем, че даден агент е избрал да сърфира във фейсбук. От това състояние той има опция да продължи да сърфира (80%) или да се върне към ученето (20%). Всички тези възможни ходове могат да бъдат изобразени графично и от тях да бъде изведена матрица на преходите. Възможен изглед на подобен граф от състояние и матрица на преходите е показана на фигура 2.

[1] „The future is independent of the past given the present“



Фигура 2 : Maël F. (no date), Markov process and transition matrix

Допълнителната стъпка направена при процеса на машинното самообучение с подкрепа е награда при всяка стъпка. Използвайки предходния пример може да дефинираме, че преминавайки към тема 3 от конспекта ще получим +5 точки, при преминаване в състояние фейсбук (-1 точка) и при връщане обратно в клас (+5 точки). Пример за подобно поставяне на награди е показан на фигура 3.



Фигура 3 : Maël F. (no date), Markov reward process

При Марковския награден процес се дефинира и т.нар. тотална награда (фигура 4). При нея имаме поредица от моментни награди (Rt) и експоненциално намаляващ фактор гамма ($\Upsilon \in [0,1]$), чиято цел е да дисконтира настоящата стойност от бъдещата награда. Посредством тази математическа трансформация може да контролираме оценката като стойности близки до нула са оценени като близко гледащи (myopic), а тези близки до 1 като далечно гледащи (far sighted) (Silver, 2015b, pp 12-15).

Definition

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Фигура 4 :Silver (2015b, p.12)

Играейки различни епизоди и използвайки тоталната награда (G_t) сме в състояние да изберем оптималното решение за дадена задача. Освен тоталната награда за епизод имаме и функция за ползата (value function), която определя очакваната възвръщаемост от състояние (t) (Silver, 2015b, p 19). Функцията на ползата е начин да измерим колко желан би бил определен ход. Посредством уравнението на Белман имаме метод чрез който да калкулираме стойността на текущото състояние като вземаме предвид само текущата награда и възможните следващи състояния (Silver, 2015b, p 20). Това на практика означава, че можем да имаме рекурсивна функция, която е валидна до края на процеса.

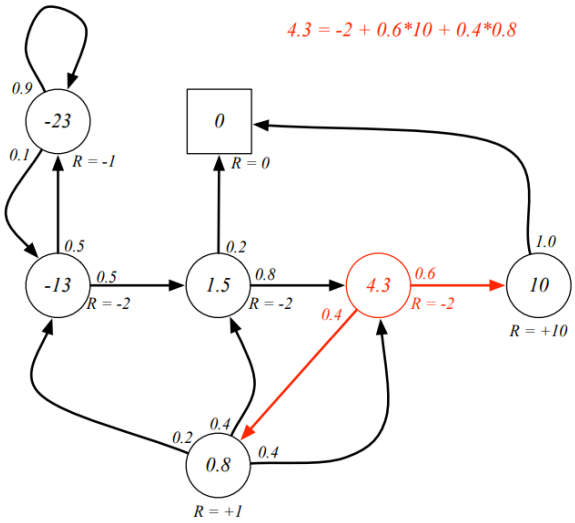
$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$

Фигура 5, уравнение на Белман (Silver, 2015b, p 20)

Уравнението на Белман представлява линейно уравнение, което може да бъде решено директно (възможно само при малки по размер процеси) или посредством редица итеративни методи като :

- Динамично програмиране
- Монте-Карло симулации
- Обучение посредством времева-разлика

Например използвайки уравнението на Белман, може да калкулираме стойността на текущото състояние (маркирана в оранжево на фигура 6) по следния начин.



Фигура 6, калкулиране на стойността на текущото състояние на Белман (Silver, 2015b, p 21)

Madani (2020) Систематизира Марковския процес за вземане на решение като Марковски процес на наградата допълнен с потенциалните възможни решения. В даден момент трябва да бъде избрано от низ от ходове, за да се осъществи придвижване между състоянията. Освен ходовете имаме и политика, която свързва състоянията към ходовете. Важна характеристика на политиките е, че те са времево независими като те зависят само от действията и състоянията. От това следва, че имаме две основни функции :

- Функция на състоянието и стойността - очакваната награда започвайки от текущото състояние следвайки дадена политика.
- Функция на ходовете и стойността – очакваната възвръщаемост започвайки от текущото състояние, предприемайки ход и следвайки дадена политика

Definition

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$$

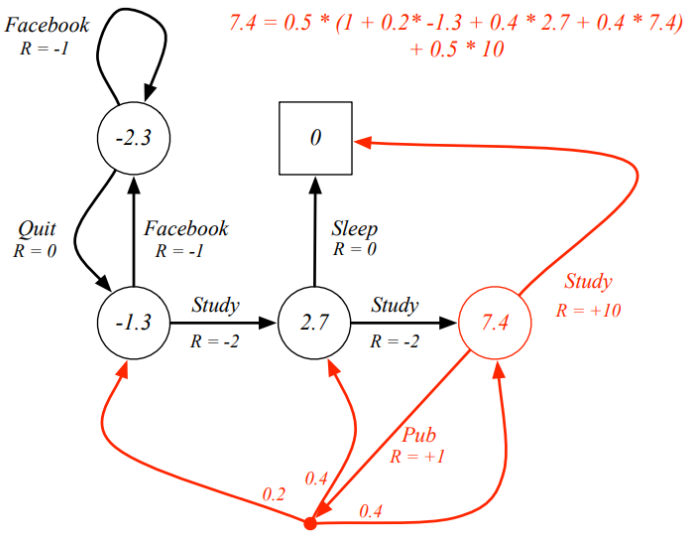
Definition

The *action-value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s, a) = \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a]$$

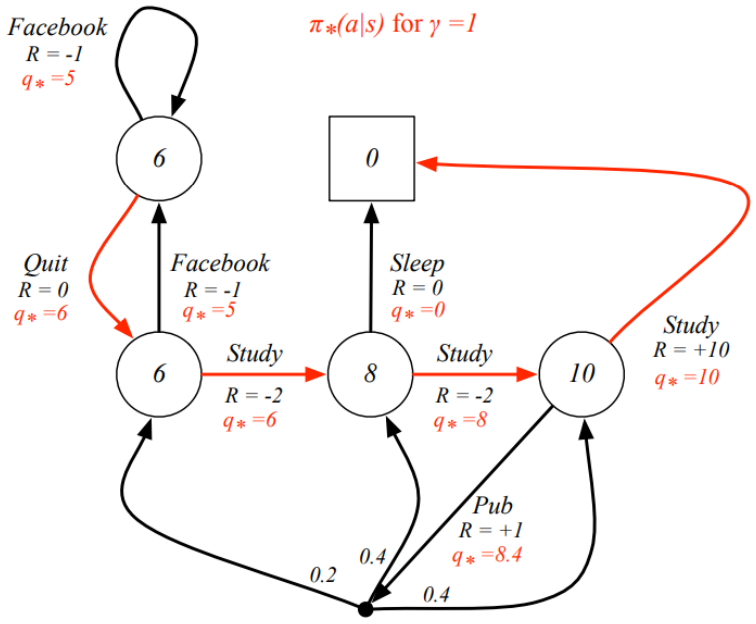
Фигура 7, функции на стойността (Silver, 2015b, p 28)

За да обобщим горното, може да разгледаме фигура 8, при която гамма (Υ = 0.1) и текущо състояние клас 3 (оранжевият кръг), ученик има 50% шанс за учене или посещение на кръчмата. От това може да калкулираме стойносста на състоянието като сумираме очакваната възвръщаемост от всеки от ходовете (уравнението в оранжево) (Madani, 2020).



Фигура 8, пример на Марковски процес за вземане на решение (Silver, 2015b, p 35)

Крайната цел на агента е да максимизира наградата като това става посредством намиране на оптимална функция на стойността. На фигура 8 е калкулирана стойността вземайки предвид всички очаквани стойности от всички възможни ходове. Докато на фигура 9 са взети предвид само действията, които носят максимална стойност. Когато знаем оптималната функция на стойността знаем и оптималната политика. От това следва, че оптимизационната задача е решена (Silver, 2015b, pp 41-42).



Фигура 9, пример на Марковски процес с оптимална политика (Silver, 2015b, p 42)

Като обобщение, Марковският процес се състои от четири основни компонента :

- Набор от състояния
- Набор от действия
- Ефект от действията
- Момента стойност от действията

Заключение

Настоящата разработка разглежда малка част от науката за вземане на решения посредством машинно обучение с подкрепа. В есето са разгледани базовите терминологии, дефинирана е взаимовръзката между средата и агента като е разгледана и една от дилемите експлоатация срещу разглеждане. В последната част е направен кратък преглед на математическо описание на взаимодействието между средата и агента посредством Марковски процес за вземане на решение.

Източници

Kanade, V. (2022) *Narrow AI vs. general AI vs. Super Ai: Key comparisons*, *Spiceworks*. Available at: https://www.spiceworks.com/tech/artificial-intelligence/articles/narrow-general-super-ai-difference/ (Accessed: February 12, 2023).

Silver, D. *et al.* (2021) "Reward is enough," *Artificial Intelligence*, 299, p. 103535. Available at: https://doi.org/10.1016/j.artint.2021.103535. (Accessed: February 12, 2023).

Sutton, R.S. and Barto, A. (2018) *Reinforcement learning: An introduction (2nd edn.)*. Cambridge, Massachusetts ; London, England: The MIT Press.

Anwar, A. (2021) *Basic terminologies of reinforcement learning*, *Medium*. Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/basic-terminology-reinforcement-learning-2357fd5f0e51 (Accessed: February 14, 2023).

Silver D. (2015a) RL course by David Silver - Lecture 1: Introduction to reinforcement learning. Available at: https://www.davidsilver.uk/wp-content/uploads/2020/03/intro_RL.pdf (Accessed: February 14, 2023).

Silver D. (2015b) RL course by David Silver - Lecture 2: Markov Decision Processes. Available at: https://www.davidsilver.uk/wp-content/uploads/2020/03/MDP.pdf (Accessed: February 14, 2023).

Mnih, V. *et al.* (2013) *Playing Atari with deep reinforcement learning*, *arXiv.org*. Available at: https://arxiv.org/abs/1312.5602 (Accessed: February 14, 2023).

Madani, A.I. (2020) *Markov decision processes simplified*, *Medium*. Level Up Coding. Available at: https://levelup.gitconnected.com/markov-decision-processes-simplified-f5f8d37ab70f (Accessed: February 14, 2023).

Maël F (no date), *Markov decision process* -. Available at: https://maelfabien.github.io/rl/RL_2/#markov-process-mp (Accessed: February 14, 2023).

Parkinson, A. (2019) *The epsilon-greedy algorithm for reinforcement learning*, *Medium*. Analytics Vidhya. Available at: https://medium.com/analytics-

vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870 (Accessed: February 14, 2023).

Chinnamgari, D.S.K. (no date) *R machine learning projects, O'Reilly Online Learning*. Packt Publishing. Available at: https://www.oreilly.com/library/view/r-machine-learning/9781789807943/8729e46c-242b-4e03-9063-2e9ad758b274.xhtml (Accessed: February 14, 2023).