УДК 004.7

К. С. Глинська, студент
Н. С. Костюкова, доцент
Донецький національний технічний університет, Україна
katerinaglinskaya96@gmail.com

Дослідження алгоритмів навчання штучного інтелекту в комп'ютерних іграх

Розглянуто можливості впровадження різних топологій й конфігурацій нейронних мереж в ігрових додатках для підсилення штучного інтелекту. Досліджено вплив параметрів нейронних мереж на їх здатність до навчання. Проведено порівняння декількох моделей нейронних мереж за критеріями точності, часу, перемог.

Ключові слова: нейронні мережі, штучний інтелект, ігровий додаток, MLP, CNN, GAN, Python.

DOI: 10.31474/1996-1588-2018-2-27-64-71

Вступ

Поняття ігрового штучного інтелекту використовується для позначення набору програмних методик для створення ілюзії інтелекту в поведінці персонажів, керованих комп'ютером, в комп'ютерних іграх. Істоту, що втілює штучний інтелект, називають агентом або комп'ютерним гравцем (NPC, non-playing character). За рахунок штучного інтелекту вдосконалюються такі аспекти ігрових додатків, як найпростіші поведінкові реакції персонажів (підбір предметів, використання об'єктів, натискання перемикачів), переміщення (обхід перешкод, стрибки, відкривання дверей), прийняття рішень (вибір одного з можливих типів поведінки, однієї з можливих операцій, визначення їх пріоритетності). Таке широке використання штучного інтелекту змушує розробників шукати та впроваджувати нові засоби його посилення, одним із яких є машинне навчання. Вивчення існуючих методів реалізації навчання штучного інтелекту на прикладі однієї гри дозволить успішно застосувати або модифікувати їх для будь-яких конкретних умов. Практика останніх років показує [1, 2], що правильно впроваджене машинне навчання в ігрових додатках дозволяє підняти їх інтелект на новий рівень. Особливим успіхом у реалізації ігрового штучного інтелекту користуються нейронні мережі як один із методів машинного навчання, тому саме на них було вирішено зосередитися в даній роботі.

Існуючі рішення у навчанні штучного інтелекту покрокових стратегічних ігор

Використання навчання в комп'ютерних іграх обґрунтовано необхідністю, оскільки у багатьох іграх звичайні алгоритми не дозволяють знайти рішення задачі за адекватний час. Наприклад, гра го має приблизно 10^{170} станів, проаналізувати які лише засобами «грубої сили» з використанням традиційних підходів неможливо [11]. Те саме можна сказати і про гру Startcraft – на сьогодні її штучний інтелект прирівнюють до гри на аматорському рівні. Автори [11] зазначають, що гра має не менш, ніж 10^{1685} можливих станів, і для реалізації такого інтелекту потрібні зовсім нові підходи до проблеми й багато досліджень.

Для реалізації штучного інтелекту розробники використовують ряд технік, які можна поділити на традиційні та нові. До перших відносять побудову дерев рішень, асоціативні правила, кінцеві автомати, експертні системи, системі нечіткої логіки, які, хоча й дозволяють імітувати розумні дії комп'ютерного гравця, багатьма авторами не вважаються повноцінним інтелектом, оскільки він не здатен до навчання. Машинне навчання включає декілька напрямків: глибинне навчання, штучні нейронні мережі, генетичні алгоритми, навчання з підкріпленням, індуктивне логічне програмування та інші. В основі машинного навчання лежить вивчення та побудова алгоритмів, що дозволяють програмі навчатись без явного програмування подальшої поведінки штучного інтелекту, покладаючись тільки на деяку наперед задану базу знань. Такі алгоритми здатні самостійно аналізувати вхідні дані та поступово вдосконалювати свій інтелект, накопичуючи досвід, і виробляти певну манеру поведінки в умовах, що постійно змінюються в ігровому світі.

Нейронні мережі дозволяють вирішити проблему слабкого штучного інтелекту, але відсутність чітких знань щодо їх впровадження у комп'ютерних іграх, складність навчання ігрових агентів стають серйозною перешкодою для розробників.

Нейронні мережі мають величезну теоретичну базу, але лише невелика її частка була випробувана на практиці створення ігрових додатків. Це пояснюється необхідністю економії пам'яті й часу обробки інформації, складністю реалізації й тестування таких розробок. Інша проблема створення нейронних мереж полягає в відсутності якихнебудь рекомендацій щодо їх застосування саме у

ігрових додатках. Нейронні мережі, як помічає Борана в [3], поки ще знаходяться на стадії «ембріонального розвитку».

Проведення комплексного аналізу різних типів мереж, порівняння функціонування різних конфігурацій в умовах певної гри могло би допомогти у виявленні певної закономірності для подальшого й більш свідомого використання нейронних мереж в ігрових додатках.

Покрокові стратегічні ігри мають багато загальних рис, що, враховуючи універсальність нейронних мереж як засобу навчання, дозволяє застосувати будь-яку одну гру для дослідження технології й подальшого її впровадження в інших подібних додатках.

Постановка задачі

Гра, що обирається для дослідження, має бути досить складною, щоб мати достатню підставу і поле для дослідження різних можливостей машинного навчання. В той же час треба враховувати обмеженість ресурсів, що будуть застосовані для розробки та дослідження. З цих міркувань в якості ігрового додатку для дослідження було обрано гру шахи. Вона досить складна для традиційних підходів, має великий ігровий простір для досліджень й є типовою стратегічною покроковою грою, що дозволить застосувати отриманні знання в інших ігрових додатках, як було зроблено у випадку з AlphaZero [4].

В рамках даної роботи розв'язується задача дослідження можливостей навчання ігрового штучного інтелекту типової стратегічної покрокової гри — шахи. У якості засобу навчання обрано апарат штучних нейронних мереж. З метою дослідження аналізуються реалізації штучного інтелекту комп'ютерного персонажа гри "Шахи" з використанням різних методів організації нейронної мережі, що дозволить виконати їх порівняння й обрати оптимальні для заданих задач параметри навчання, що матиме істотний вплив на якість ігрового процесу як у розробленій грі, так і у майбутніх розробках цього жанру, зокрема, у покрокових стратегічних іграх для двох суперників.

Аналіз можливостей використання нейронних мереж для вирішення задачі

Для задачі, що вирішується, вхідними даними мережі є поточний стан шахової дошки, на виході можна отримувати або новий стан дошки, або конкретний хід.

Для реалізації штучного інтелекту стратегічної гри можна застосувати щонайменше три види (топології) нейронних мереж:

1. Багатошаровий персептрон (MLP, multilayer perceptron).

2. Згорточна нейронна мережа (CNN, convolutional neural networks).

3. Генеративно-змагальна мережа (GAN, generative adversarial network).

Багатошаровий персептрон є одним з найбільш вивчених та простих варіантів реалізації нейронної мережі. Навчання, як правило, здійснюється методом зворотного поширення помилки (backpropagation, модифікація методу градієнтного спуску) та методом навчання в вчителем, тобто ми повинні надати для навчання набір пар «вхідний вектор – правильний вихід». Вхідний вектор відправиться на вхід мережі, послідовно будуть розраховані стани всіх проміжних нейронів, і на виході обраховується вихідний вектор, який ми і порівняємо з правильним. Розбіжність дасть нам помилку, яку можна поширити назад по зв'язкам мережі, обчислити внесок в підсумкову помилку кожного нейрона і скорегувати його ваги, щоб її зменшити. Після багатокрокового повторення помилка має понижатися, а мережа видавати точніші відповіді.

Для реалізації такої топології потрібні навчальні матеріали – датасети – набори вхідних даних та очікуваних (правильних) результатів, за допомогою яких мережа буде навчатись. Завдяки довгій історії розвитку шахової гри, існують мільйони записів шахових партій, які можна застосувати для навчання, включаючи партії кращих шахових гравців протягом останніх десятків років. Для будь-якої іншої гри датасетів може не існувати. В цьому випадку гра може самостійно створити необхідну базу даних засобом гри "сама із собою", поступово накопичуючи кращі ходи й використовуючи результат партій для подальшого навчання (рис.1).

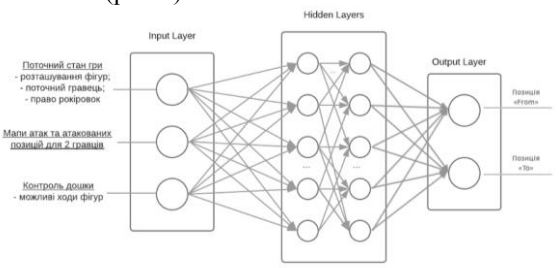


Рисунок 1 – Топологія "Багатошаровий персептрон" для гри в шахи
(авторський рисунок)

Згорточна нейронна мережа пропонує зовсім іншу архітектуру, яка дозволяє виділяти окремі риси вхідних даних (зображень), переходити від їх конкретних особливостей до більш абстрактних деталей, складати ієрархію рис, тобто фільтрувати незначні деталі і виділяти істотне (рис. 1). Ці здібності згорточної нейронної мережі ми і маємо перенести до вирішення проблем ігрового процесу, адже саме ці задачі зазвичай вирішує гравець стратегічної гри під час вибору кращого ходу. Згорточна мережа має приймати на вхід зображення, але шахова дошка і є в певному

виді зображенням (або матрицею 8 на 8), як і будь-яка інша стратегічна гра (рис.2).

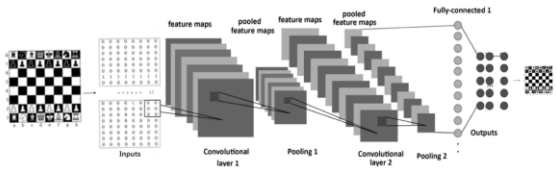


Рисунок 2 – Топологія "Нейронна мережа згортки" для гри в шахи (авторський рисунок на основі [2])

Генеративно-змагальна нейронна мережа базується на комбінації двох нейронних мереж (які, в свою чергу, можуть мати різні архітектури, зокрема, персептрон або згорточна мережа), одна з яких (генератор) вчиться генерувати якісь дані

(в нашому випадку правильні ходи), а друга (дискримінатор) – відрізняти правильні ходи від помилкових. Таким чином, між двома мережами виникає антагоністична гра, мережі навчаються по черзі, намагаючись перевершити одна одну: генератор має навчитись пропонувати такі ходи, щоб дискримінатор не зміг відрізнити їх від зразкових (правильних). Дискримінатор, в свою чергу, вчиться відрізняти гарні ходи від помилкових. Завдяки методу зворотного поширення помилки результат роботи дискримінатора повертається генератору, щоб він міг покращити свою роботу. У процесі спільного конкурентного навчання, якщо система досить збалансована, досягається мінімаксний стан рівноваги, в якому обидві мережі значно поліпшили свою якість (рис.3).

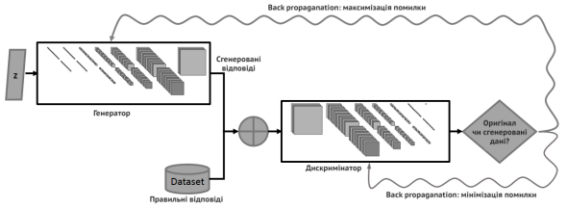


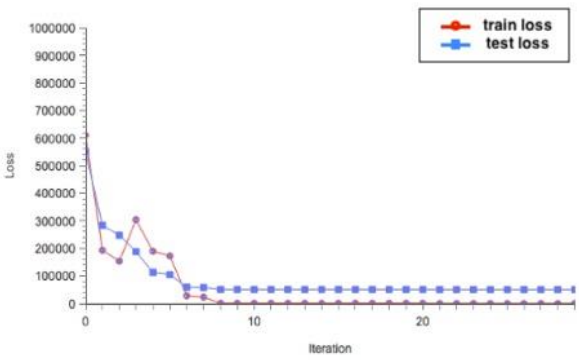
Рисунок 3 – Топологія "Генеративно-змагальна мережа" (авторський переклад на основі [12])

Основною властивістю нейронної мережі є її здатність до узагальнення: навчившись грати в шахи на певному тренувальному наборі, вона має правильно реагувати навіть на незнайомі їй позиції й кроки суперника. Однак частою проблемою при навчанні нейронної мережі є її схильність до перенавчання, тобто ситуація, коли мережа лише запам'ятовує кращі кроки для певних умов й втрачає здатність до узагальнення [6]. Існує ряд засобів запобігти перенавчанню мережі, зокрема, використання методу dropout як одного з методів регулярізації [3]. Цей метод також дозволяє прискорити навчання мережі, однак за деяких умов мережа може втратити здатність до навчання через надмірне спрощення її структури. Параметри для навчання такої нейронної мережі слід підбирати експериментальним шляхом.

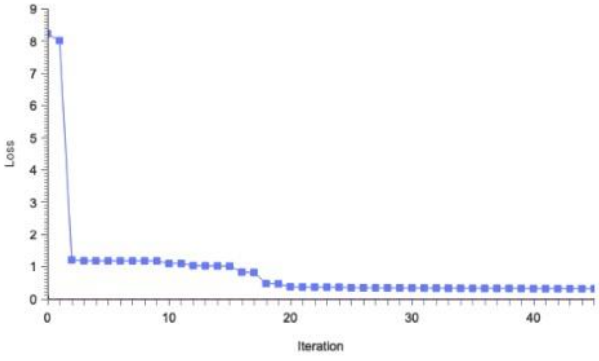
Перший етап такого підбору реалізовано шляхом моделювання спрощеної копії мережі (з меншим числом нейронів й меншим тренувальним датасетом) для того, щоб побачити тенденцію зміни похибки залежно від параметрів навчання. Як середовище моделювання була використана програма NeuralDesigner [7], що дозволяє перевірити ряд гіпотез щодо ефективності тієї чи іншої конфігурації нейронної мережі на невеликому наборі даних ще під час проектування системи.

Зокрема, на рис. 4 наведені результати моделювання багатошарового персептрону з різною кількістю нейронів, шарів, алгоритмами активації нейронів та оптимізації. Як бачимо, похибка зменшується в обох випадках, однак на рис. 4 (а) зменшення похибки зупиняється на високому рівні, й можна припустити, що дана модель досягла максимальної точності й подальше навчання не покращить результати.

Модель нейронної мережі з рис. 4 (б) також зупиняє навчання, але в цьому випадку похибка значно менша. В першому випадку було обрано занадто просту топологію з малим числом нейронів. В другому випадку нейронна мережа мала 3 внутрішніх шари з 256 нейронами на кожному з них. Подальше збільшення кількості нейронів дозволили зменшити похибку до 0.2, після чого моделювання мережі даної топології було припинено.



а)



б)

Рисунок 4 – Попереднє моделювання мережі

Таким чином, не застосовуючи повний процес навчання нейронної мережі, було перевірено декілька конфігурацій нейронних мереж, що дозволило дійти таких висновків:

1. Велика кількість шарів з невеликою кількістю нейронів не підсилює інтелект нейронної мережі.

2. Збільшення кількості нейронів в шарі до певного моменту зменшує помилку, але після цього не впливає на навчання мережі.

3. Кількість нейронів на внутрішніх шарах має бути в декілька разів більше, ніж кількість вхідних нейронів (вхідних даних).

Вибір алгоритмів для навчання

Розглянемо детальніше використані алгоритми навчання й роботи нейронних мереж. Загальний алгоритм роботи з будь-якою нейронною мережею представлений на рис. 5.

Рисунок 5 – Загальний алгоритм підготовки мережі (авторський рисунок)

Навчання MLP й CNN з вчителем майже не відрізняється. В якості тренувальних даних застосовуються шахові партії гравців з рейтингом ELO 2000+ [8]. Це має забезпечити гру розроблених моделей на рівні не меншим за кандидата у майстра спорту з шах. Ми мали можливість використати й партії сильніших гравців, але ця робота націлена на порівняння й пошук кращої топології й конфігурації мережі, тобто порівняння розроблених моделей між собою, а не існуючими гравцями.

Мережі отримують тренувальні набори даних (позиції на полі) разом з правильною відповіддю (крок, зроблений сильним гравцем). Після певної кількості ітерацій ваги мереж оновлюються, щоб розбіжність між її результатом й очікуваною відповіддю була зменшена.

Навчання генеративно-змагальної мережі дещо відрізняється. Обидві мережі (генератор й дискримінатор) навчаються по черзі, коли навчається дискримінатор, ваги генератора заблоковані, тобто не можуть змінюватись, й навпаки. Після навчання мережі необхідність в дискримінаторі відпадає – він потрібен лише на етапі навчання, для роботи шахового інтелекту використовується лише генератор. Обидві нейронні мережі створюються, компілюються, й можуть бути збережені незалежно.

Для розробки програмного продукту було обрано мову програмування Python версії 3.6 та бібліотеку для машинного навчання TensorFlow 1.10. TensorFlow використовується як бекенд при роботі з фреймворком Keras.

Відсутність графічної відеокарти Nvidia не дозволяє запускати навчання й використання нейронної мережі на GPU. Це варто враховувати при порівнянні розробленої програми з аналогічними розробками. Наприклад, краща на сьогодні реалізація нейронної шахової мережі AlphaZero використовувала при навчанні 64 GPU та 19 CPU [9], а загальна її вартість за приблизними оцінками досягла 25 мільйонів доларів. З іншого боку, дана робота дозволить довести, що навчання нейронної мережі можливе навіть на звичайному користувальницькому обладнанні.

Результати й аналіз

Тестування нейронних мереж було виконано в 3 етапи:

1. Пошук оптимальних параметрів нейронних мереж, тобто порівняння кожної топології з тією ж самою топологією та різними параметрами навчання моделі.

2. Порівняння отриманих топологій між собою за критерієм точності й часу навчання.

3. Порівняння розроблених моделей в процесі гри одна з одною, з іншими розробками або на сторонніх тестових наборах.

Тестування мереж з метою пошуку кращих параметрів проводилось невеликими етапами або 1-2 години, або до 20-30 епох залежно від виду випробування. Звичайно, за цей час мережа не досягала великої точності, але це дозволяло оцінити вплив параметрів на швидкість збіжності й рухатись в правильному напрямку.

Оцінка й порівняння нейронних мереж були виконані за критерієм точності їх роботи, тобто за

здатністю моделі запропонувати той самий хід, що робить майстер спорту з шах в партіях, що аналізуються. Окремо був використаний вебресурс [10], що дозволяє виконати аналіз шахової партії (записаної в форматі PGN) за допомогою движка Stockfish (до недавніх пір – сильнішого шахового движка).

На рис. 6 (а) наведено приклад пошуку оптимальної (з точки зору точності) кількості нейронів на кожному внутрішньому шарі в MLP. На рисунку похибка мережі на тренувальному наборі безперервно падає протягом збільшення кількості нейронів, однак на тестовому наборі в певний момент вона навіть збільшується. Це яскравий приклад перенавчання мережі. Можна зрозуміти, що кількість нейронів менша за 500 не дозволить мережі досягнути гарної точності роботи, і в цей же час збільшуючи кількість нейронів до 1250 і більше, ми ризикуємо зменшити здатність мережі до узагальнення.

На рис. 6 (б) наведено процес навчання тієї ж мережі, але цього разу досліджується вплив кількості епох на результат роботи мережі. Побудова такого графіку залежності під час навчання дозволяє своєчасно виявити перенавчання мережі. Навчання слід припиняти, коли похибка перестає зменшуватись. В даному випадку точність на тренувальному й тестовому наборах майже не відрізняється, тому можна припустити, що потужність мережі й обсяг навчальних даних обрано правильно.

На рис. 6 (в) виконується пошук оптимального значення параметру dropout. Зазвичай в багатьох задачах рекомендується використовувати значення 0.2 (20%), одна в нашому випадку 30% дало кращу точність. Dropout був використаний у всіх моделях нейронних мереж.

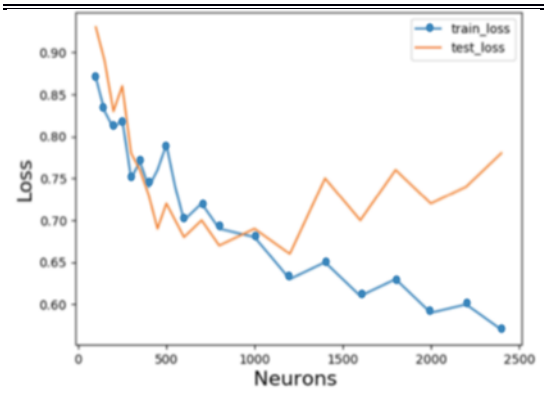
Схожу роль в навчанні нейронних мереж грає регулярізація. У машинному навчанні це метод додавання деякої додаткової інформації до вхідних умов з метою запобігти перенавчанню. Перенавчання в більшості випадків виявляється в тому, що в многочленах, які утворюються, занадто великі коефіцієнти. Тому ця додаткова інформація часто має вигляд штрафу за складність моделі (за великі коефіцієнти). В нейронних мережах для цього застосовується L2-регулярізація, або регулярізація Тихонова (ridge regression, weight decay або Tikhonov regularization). Якщо коефіцієнт регулярізації занадто малий, то ефект від регуляризації буде мізерний, якщо ж занадто великий – модель обнулить все ваги. На рис. 5 (г) наведено значення точності навчання мережі для різних значень коефіцієнту регулярізації.

Окремо треба звернути увагу на розмір партії (batch_size). Розмір партії в ітераційному градієнтному спуску – це кількість шаблонів, показаних мережі перед оновленням ваг. Це також засіб оптимізації тренування мережі, що визначає, скільки шаблонів потрібно читати одночасно та зберігати в пам'яті. При занадто великому значенні batch_size у системі може просто не вистачити пам'яті, при маленькому значенні мережа не буде навчатись, адже даних буде не вистачати для розумного оновлення ваг. Згорточні мережі (CNN) більш чутливі до цього параметру, ніж персептрон, однак для різних задач оптимальнішими можуть виявитись різні значення, тому кращим засобом виявити його буде запустити навчання декількох мереж з різними batch_size й простежити зміну точності роботи. В нашому випадку були здійснені експерименти з batch_size від 10 до 50 з кроком 10 для різних топологій мереж для 50 епох. Більші значення batch_size показали кращі результати. Оскільки шахові датасети мають компактні розміри (у порівнянні, наприклад, з обробкою зображень), проблем з пам'яттю виявлено не було, однак було рішено не збільшувати розмір партії далі (більше 50 елементів). Результати експериментів наведені на рис. 6 (ґ).

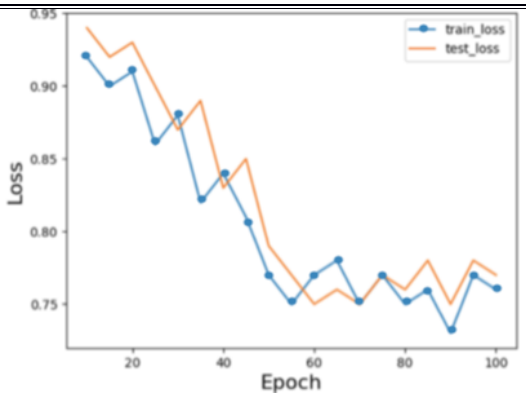
Після визначення параметрів можна перейти до порівняння мереж між собою. На рис. 6 (д) наведені графіки зростання точності навчання трьох топологій мереж в процесі збільшення епох. Можна помітити, що точність навчання MLP зростає скоріше за всіх – це пов'язано з тим, що модель персептрона найпростіша й має менш параметрів, які слід оптимізувати. Однак проста структура має очевидний недолік – навчання MLP зупиняється на точності приблизно 60%. В цей же час навчання згорточної мережі хоча і йде повільніше, однак складніша структура дозволяє досягти більшої точності. Генеративно-змагальна мережа, як і можна було очікувати, не виявила здатності до швидкого навчання. Слід пам'ятати, що вона складається з двох незалежних мереж, має набагато більше параметрів, й вимагає більше часу для оптимізації. На графіку можна побачити, що похибка навчання зменшується, й можливо подальші експерименти з моделлю дозволять прискорити цей процес.

Після незалежної оцінки кожної мережі слід приступити до ігор між різними моделями, щоб перевірити їх в «польових умовах». На рис. 6 (е) наведено аналіз шахової партії між моделями MLP й CNN після 100 епох навчання. За графіком 5 (д) можна очікувати приблизно однаковий рівень гри обох мереж на цьому етапі навчання. Білі (CNN) перемагають, однак аналіз сторонньою програмою показує, що обидві сторони грають непогано, обираючи майже завжди кращі або гарні кроки, не роблячи помилок.

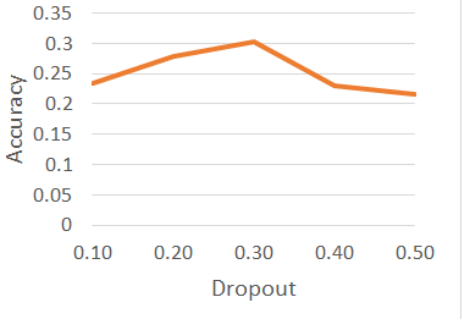
На рис. 6 (є) наведено результат гри тих самих моделей, але вже наприкінці навчання. CNN знов грає за білих і одержує перемогу. Різниця між гравцями помітніша, однак можна сказати, що обидві сторони грають розумно, роблячи гарні ходи.



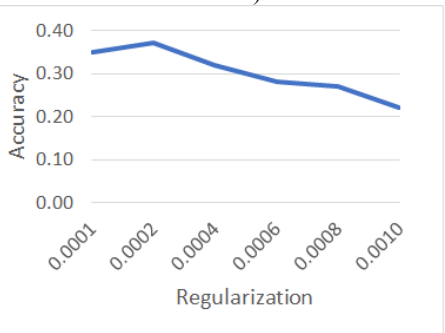
а)



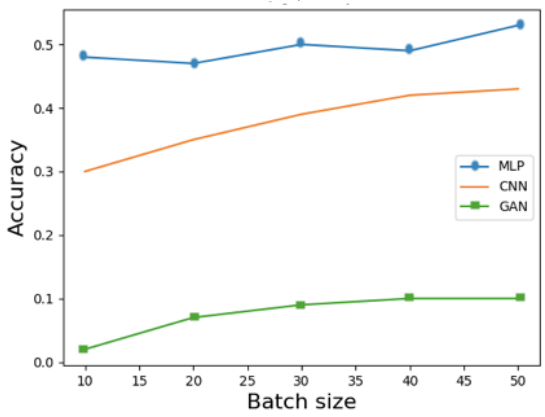
б)



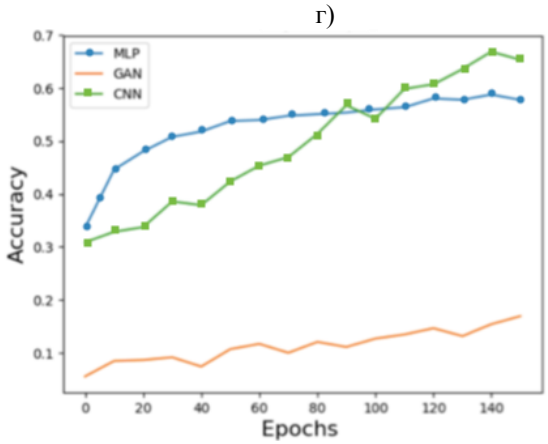
в)



г)



г)



д)

Сила	Белые	Черные
Отлично	40	36
Хорошо	4	6
Неточность (?!)	1	1
Ошибка (?)	0	0
Зевок (??)	0	0
Форсировано	0	1
Лучший ход	81,4%	66,7%
Ср. Разница	0,06	0,10

е)

Сила	Белые	Черные
Отлично	15	12
Хорошо	3	5
Неточность (?!)	1	1
Ошибка (?)	0	0
Зевок (??)	0	0
Форсировано	0	0
Лучший ход	64,3%	35,7%
Ср. Разница	0,09	0,14

є)

Рисунок 6 – Результати тестування

Висновки

В процесі роботи були розроблені 3 моделі нейронних мереж для гри в шахи.

За результатами порівняння, кращими за всіма критеріями виявились моделі MLP й CNN. Персептрон трохи поступається згорточній мережі за точністю, але потребує менш часу на навчання, використовує менше пам'яті й показує кращу швидкодію. GAN й модель, заснована на навчанні з підкріпленням, навчаються занадто повільно, щоб суперничати з іншими движками, але також показують прогрес у процесі навчання.

При порівняння з іншими шаховими движками моделі CNN й MLP гарно показали себе, роблячи розумні, а часто й кращі в певних умовах кроки. Однак слід розуміти, що вони поки що відстають від багатьох аналогічних додатків й можуть бути покращені. Зокрема, слід зосередитись на згорточній мережі як більш перспективній й продовжити її навчання протягом більшої кількості епох, поки помилка продовжує падати. Використання GPU дозволить скоротити час на навчання, а ряд додаткових прийомів, таких як нормалізація партії, крос-валідація, Grid Search, можуть підвищити точність навчання. Також слід продовжити експерименти з GAN, застосувавши для навчання потужніші процесори, оскільки ця топологія більш вимоглива до ресурсів, але й показує гарні результати в багатьох інших галузях.

Ряд експериментів дозволив виявити оптимальні параметри навчання, переваги й недоліки кожної топології. Схожість стратегічних ігор між собою дозволяє використати отримані дані в інших комп'ютерних іграх близького типу, застосувати нейронні мережі для реалізації штучного інтелекту схожих ігрових агентів.

Список літератури

1. Millington I., Funge J. Artificial Intelligence for Games, Second Edition. – Elsevier Inc., 2009. – 895 p.
2. Yannakakis G., Togelius J. Artificial Intelligence and Games. – Springer, 2018. – 359 p.
3. J. Borana Applications of Artificial Intelligence. – Int. Conf. Emerg. Technol. Eng. Biomed. Manag. Sci., no. March, pp. 64–67, 2016.
4. Игры кончились: AlphaGo займется решением реальных мировых проблем [Електронний ресурс]. – Режим доступу: https://hi-news.ru/computers/igry-konchilis-alphago-zajmetsya-resheniem-realnyx-mirovyx-problem.html.
5. A. Cîmpean Lucrare De Diplomă "Machine Learning-based gameplay". – pp. 1–57, 2012.
6. Memorizing is not learning [Електронний ресурс]. – Режим доступу: https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42
7. Neural Designer [Електронний ресурс]. – Режим доступу: https://www.neuraldesigner.com/.
8. FICS Games Database [Електронний ресурс]. – Режим доступу: https://www.ficsgames.org/download.html.
9. How much did AlphaGo Zero cost [Електронний ресурс]. – Режим доступу: https://www.yuzeh.com/data/agz-cost.html.
10. Аналіз шахової партії [Електронний ресурс]. – Режим доступу: https://www.chess.com/ru/analysis-board-editor.
11. Usunier N. Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks / N. Usunier, G. Synnaeve, Z. Lin, S. Chintala // ICLR-2017. – pp. 1–18.
12. Oudeyer P.-Y. Autonomous development and learning in artificial intelligence and robotics: Scaling up deep learning to human-like learning / P.-Y. Oudeyer // Behav. Brain Sci. – 2017. – №40 (275). – pp. 1–13.

References

1. Millington, I., Funge, J. (2009), Artificial Intelligence for Games, Second Edition, Elsevier Inc., 895 p.
2. Yannakakis, G., Togelius, J. (2018), Artificial Intelligence and Games, Springer, 359 p.
3. Borana, J. (2016), Applications of Artificial Intelligence, Int. Conf. Emerg. Technol. Eng. Biomed. Manag. Sci., no. March, pp. 64–67.
4. Games are over: AlphaGo will solve real world problems [Igry konchilis': AlphaGo zaĭmetsya resheniyem real'nykh mirovykh problem], vailable at https://hi-news.ru/computers/igry-konchilis-alphago-zajmetsya-resheniem-realnyx-mirovyx-problem.html.
5. Cîmpean Lucrare De Diplomă, A. (2012), Machine Learning-based gameplay, pp. 1–57.
6. Memorizing is not learning, available at https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42
7. Neural Designer, available at https://www.neuraldesigner.com/.
8. FICS Games Database, available at https://www.ficsgames.org/download.html.
9. How much did AlphaGo Zero cost, available at https://www.yuzeh.com/data/agz-cost.html.
10. Chess Party Analysis [Analiz shakhovoyi partiyi] , available at https://www.chess.com/ru/analysis-board-editor.

11. Usunier, N., Synnaeve, G., Lin, Z., Chintala, S. (2017), Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Task, ICLR-2017, pp. 1–18.
12. Oudeyer, P.-Y. (2017), Autonomous development and learning in artificial intelligence and robotics: Scaling up deep learning to human-like learning, Behav. Brain Sci., №40 (275), pp. 1–13.

Надійшла до редакції 10.12.2018

К.С. ГЛИНСКАЯ, Н.С. КОСТЮКОВА
Донецкий национальный технический университет, г. Покровск, Украина

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБУЧЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В КОМПЬЮТЕРНЫХ ИГРАХ

Рассмотрены возможности внедрения различных топологий и конфигураций нейронных сетей в игровых приложениях для усиления искусственного интеллекта. Исследовано влияние параметров нейронных сетей на их способность к обучению. Проведено сравнение нескольких моделей нейронных сетей по критериям точности, времени, побед.

Ключевые слова: нейронные сети, искусственный интеллект, игровой приложение, MLP, CNN, GAN, Python.

K. HLYNSKA, N. KOSTYUKOVA
Donetsk National Technical University, Pokrovsk, Ukraine

RESEARCH OF ALGORITHMS OF LEARNING ARTIFICIAL INTELLIGENCE IN COMPUTER GAMES

The work is devoted to different approaches to the implementation of the process of learning game artificial intelligence by means of neural networks in order to enhance the intelligence of computer agents and the quality of the game process.

The object of research is algorithms and methods for the implementation of artificial intelligence. The subject of the study is neural networks as a way of machine learning that solves the problem of creating artificial intelligence in situations where traditional algorithms and manual programming of actions are not possible.

The use of neural networks makes it possible to simulate artificial intelligence, capable of self-organization, learning and self-learning, adapting behaviour to new conditions, memorizing and analysing input data. Today, the operation of neural networks is a promising issue, as the effective use of technology in modern gaming applications suggests, but there are still many difficulties that need to be solved, and their use in gaming applications is poorly understood and limited by obstacles that prevent the use of all the benefits of technology in this field. Therefore, despite the large theoretical basis, neural networks remain a poorly-studied and actual problem.

The study of existing means of constructing and training a neural network on the example of a certain strategic game will determine the optimal parameters of its functioning for further use in the gaming industry. The game application to be developed should not be inferior to the human player in making decisions and strategy games, modelling the behaviour and thinking of the person, satisfying the requirements of modern users to the level of artificial intelligence and the likelihood of the game.

Keywords: artificial intelligence, teaching methods, neural networks, game application, machine learning, strategic games, learning algorithms.