

УДК 004.8

Дорогий Я.Ю., к.т.н.,
orcid.org/0000-0003-3848-9852,
Свириденко О.А.,
orcid.org/0000-0003-0241-3485

ТАКТИЧНИЙ ШТУЧНИЙ ІНТЕЛЕКТ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СТРАТЕГІЇ В РЕАЛЬНОМУ ЧАСІ

Національний технічний університету України «Київський політехнічний інститут ім. Ігоря Сікорського»

argusyk@gmail.com

Вступ

В стратегіях реального часу проблема побудови якісного штучного інтелекту для управління військами та загальним напрямом розвитку є проблемою для науковців ще з моменту зародження жанру. Для комерційних проектів використання сучасних підходів для створення штучного інтелекту – занадто дорогий варіант, тому найчастіше використовується написаний статичний алгоритм, для ігрової індустрії використовують термін «заскриптований». Проблема цих алгоритмів полягає в простоті реагування та передбачуваності їх дій, тому гравцям стає швидко нецікаво грати проти комп'ютера. Протилежною до попередньої є проблема агентів, що використовують попередньо визначений алгоритм, але не дивлячись на це, показують хороші результати у боротьбі гравцями-людьми через високу швидкість дій та доступ до внутрішньої інформації гри – наприклад, місцезнаходження об'єктів супротивника, які сховані туманом війни. Все це викликає негатив у користувачів ігрових продуктів, який можна уникнути використовуючи більш просунуті технології та алгоритми для створення штучного інтелекту.

Постановка проблеми

Останнім часом активно розвивається ринок мобільних ігор, і для ефективної роботи на цьому ринку потрібно створювати рішення і іграх, які дозволять, використовуючи менші можливості мобільних пристроїв, отримати результат, що по якості не буде поступатись результатам на ПК. Якщо раніше основною проблемою

була графіка, то тепер серед користувачів виникає потреба в якісному штучному інтелекті. Для ігор в стратегічному жанрі цікавий і потужний штучний інтелект завжди був проблемою, а на мобільних пристроях з'явилась додаткова проблема з обмеженими можливостями системи. Особливо явно ця проблема виражається в тактичній складовій, коли потрібно керувати великою кількістю юнітів, що знаходяться в бою з супротивником. Тому основною задачею дослідження, описаного в статті, є побудова гнучкого, результативного тактичного штучного інтелекту, що буде орієнтованим на мобільні пристрої та, відповідно, обмежені можливості фізичних ресурсів системи.

Використання нейронних мереж для створення агентів тактичного управління

Загалом стратегії реального часу є моделюванням реальних подій під час військових конфліктів у дещо спрощеному середовищі. Через це для створення штучного інтелекту використовують декілька умовних модулів, які досить сильно відрізняються за своїми цілями і, відповідно, алгоритмами та методами реалізації. Агенти вищого рівня – стратегічний штучний інтелект – використовується для прийняття стратегічних цілей. Агенти розвідки та визначення планів супротивника використовуються для коригування дій стратегічного штучного інтелекту і прогнозування можливих дій супротивника. І останніми є агенти мікроменеджменту – тактичного штучного інтелекту. Їх-

нім завданням є управління військами і максимізація ефективності кожної окремої бойової одиниці.

Для створення агентів тактичного штучного інтелекту дослідники та комерційні компанії використовують різні алгоритми, однак більшість сучасних продуктів або дослідів концентрується на використанні нейронних мереж.

Дослідники серед різних видів нейронних мереж в першу чергу виділяють згорткові – вони показали себе найефективнішими з точки зору боротьби з ботами, не дивлячись на відносну повільність роботи згорткових нейронних мереж [1]. Також такий тип нейронних мереж показав свою ефективність у прийнятті рішень щодо визначення переможця у конкретній ситуації.

Серед комерційних проектів можна виділити гру Blitzkreig 3 [3], де використано нейронну мережу як агент для управління бойовими одиницями на полі. Агент здатен керувати кожною окремою бойовою одиницею, при цьому використання кожної одиниці є максимально ефективним – при недостатній кількості одиниць здоров'я агент відводить бойову одиницю з передової за укриття і надсилає одиниці підтримки для відновлення. Також, агент здатен прибирати одиниці з лінію вогню для мінімізації втрат та економії ресурсів, оскільки ціна створення нової одиниці вища, ніж ціна відновлення.

Нейронні мережі дозволяють створити штучний інтелект, який є досить гнучким у прийнятті рішень і є достойним супротивником навіть гравцям-людям.

Окрім нейронних мереж для створення штучного інтелекту, який буде керувати тактичними діями, також використовують наступні підходи:

- Пошук за ігровим деревом;
- Баєсова модель;
- Рішення на основі прецедентів.

Баєсова модель, яка представляє собою набір випадкових змінних та залежностей між ними за допомогою орієнтова-

ного ациклічного графу, використовувалась для визначення напрямку руху юнітів [4].

З використанням підходу пошуку за ігровим деревом дослідниками створено систему, яка, аналізуючи відомі стратегії, шукала рівновагу Неша – за умов цієї рівноваги супротивник не може отримати переваг (збільшити виграш) змінивши стратегію в односторонньому порядку [5]. Для дослідження обрано лише тактичну складову, а також виключено мікроменеджмент кожної окремої бойової одиниці – управління відбувалося групами і концентрувалось на переміщенні. Як результат, система змогла перемогти бота у грі практично у всіх ситуаціях, адже на кожну конкретно визначену розробниками стратегію ігрового бота система відповідала покращеною стратегією.

Прийняття рішень на основі прецедентів в широкому сенсі є вибором дії на основі вже відомих послідовностей. Створений у 2009 році дослідниками модуль для гри Warcraft [6] зміг ефективно показати себе у мікроконтролі в битвах з вбудованим штучним інтелектом та стати ефективним помічником гравцю, забезпечуючи ефективне мікроуправління військами, тим самим надаючи гравцю можливість сконцентруватись на високо-рівневому плануванні – стратегічному.

Підготовка тестового середувища

Для тестування обрано спрощений варіант стратегії з акцентом на тактичну складову. Кожному з гравців доступно 2 види бойових одиниць, кожна з яких є одиницею ближнього бою. В залежності від типу відрізняється кількість одиниць здоров'я, сили та броні. Кожна одиниця на полі має функції, які характеризують ціну бойової одиниці та умовний показник її ефективності. Ці дані необхідні в подальшому, оскільки для правильного аналізу ситуації на мапі бот має правильно розуміти розклад сил.

Основне завдання бота – тактичне управління групою бойових одиниць. Для спрощення роботи всі можливі дії гравців

в рамках тестового середовища було класифіковано. Загалом, не дивлячись на велику кількість варіацій, управління ігровими елементами в іграх жанру стратегій в реальному відбувається за допомогою обмеженої кількості команд. Команди можуть стосуватись бойових одиниць або будівель. Оскільки в даній роботі розглядається тільки тактична складова стратегії, класифікація наказів буде стосуватись управління бойовими одиницями. Для стратегічного управління всі накази також можна класифікувати, але наразі в цьому немає потреби.

Загалом, основними наказами для бойових одиниць будуть наступні:

- Захист позиції;
- Атака позиції;
- Рух до позиції.

Різниця між наказом «рух до позиції» та «атака позиції» полягає у реакції бойових одиниць на ворога. Якщо при першому варіанті одиниці будуть рухатись незважаючи на ворога, то в варіанті з атакою бойові юніти будуть атакувати будь кого з ворогів, що потраплять в зону контролю. На функціоналі, який будуть підтримувати бойові одиниці завдяки цим варіантам управління можна побудувати додаткові накази, які полегшать гру для гравців-людей. Наприклад, популярний в стратегіях в реальному часі наказ «патрулювання» за своєю суттю є зацикленним режимом «атаки позиції» між двома точками.

Окрім наказів у грі присутні «події» – важливі зміни в ігровому середовищі, що є наслідками наказів, однак незалежні від одного гравця. Наприклад, контакт військ з супротивником – це подія, яка викликає ряд функцій, які на неї підписані.

Ще однією особливістю тестового середовища є управління тільки тими бойовими одиницями, що є на екрані. Це дозволить гравцям бути в більш рівних умовах з ботами, а для штучного інтелекту великою перевагою буде зменшення ресурсо-затратності прийняття тактичних рішень, оскільки агент буде аналізувати тільки «активну» частину карти.

Розробка агента для прийняття рішень

Для розробки штучного інтелекту буде застосовано штучні нейронні мережі, з певною специфікою, оскільки дані, які потрібно обробляти, мають комплексну структуру, та є нетиповими для використання в мережах.

Перш за все, для штучного інтелекту важливо враховувати правила тестового середовища. Тому агент буде використовувати доступні дані з певної зони – по суті те, що бачив би гравець. Отримані дані є набором координат і характеристик військ. Для аналізу дані потрібно обробити і класифікувати.

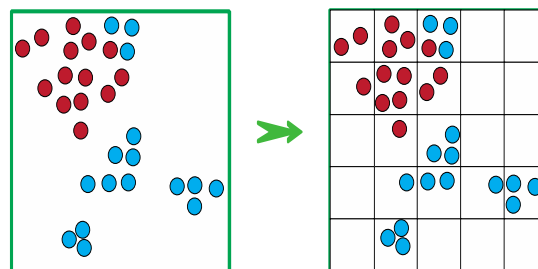


Рис. 1. Поділ території на зони

Отримані дані потрібно розбити на визначені невеликі частини. Управління – віддавання наказів – буде реалізовано для груп бойових одиниць, де кожен квадрат – окрема група (при відсутності військ накази не будуть віддаватись). Після розділу на групи умовна зона аналізу буде мати 25 під-зон, в кожній з яких буде окрема група під управлінням бота. Кожна під-зона має свій умовний номер, від 1 до 25, рахуючи зліва направо і зверху вниз.

Після попередньої обробки даних можна починати етап аналізу. Варто зауважити, що через агент буде аналізуватися ситуація та оновлюватися накази через визначені періоди часу – середній час на виконання попередньої команди. Також варто зауважити, що управління кожною окремою групою буде проводитись в рамках 9 клітинок (рис.2). Кінцева точка наказу для пересування групи буде в рамках «зони впливу» групи.

Візуалізація зон аналізу впливу і розміщення для 3 зони зображена на рис.2.

Зона для аналізу позначена червоним, сусідні зони, що разом складають зону впливу, позначені червоною пунктирною лінією.

Мету роботи штучного інтелекту можна сформулювати досить чітко – максимізація ефективності використання одиниць, що потрапляють в територію управління. Для метрики ефективності в зоні контролю агенту буде встановлено систему обрахунку втрат бойових одиниць – як супротивника, так і союзних. Основною метрикою системи буде вартість одиниці, яка буде отримуватись з відповідної функції в кожного юніта в рамках тестового середовища.

Для прийняття рішення щодо подальших дій групи для кожної окремої зони буде проводитись 2 окремі дослідження – для аналізу зон впливу та аналізу ситуації в безпосередньому місці розміщення групи. На рис.3 можна побачити, як з карти території, розділеної на зони, ми отримуємо два результуючих набори даних, що є по суті матрицями результатів аналізу обох досліджень.

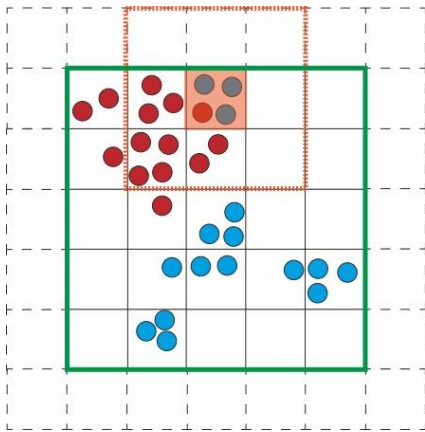


Рис. 2. Аналіз для зони номер 3

Фактичним результатами, з точки зору людини, будуть дані про можливість утримання поточної позиції виключно силами, які в ній знаходяться. Другий набір даних містить інформацію про позицію з урахуванням загальної кількості союзних та ворожих військ. Вплив – поняття умовне, оскільки під ним мається на увазі можливість переходу в зону військ сусідніх зон за одиницю часу. Величина «впливу»

залежить від кількості ворожих та союзних військ в рамках частини території, розміром 9 зон.

На рис.3 також навколо території, що виділена зеленим, додано ряд зон. Необхідність в додаткових зонах полягає у єдиному підході для аналізу оточуючих зон. Оскільки для аналізу потрібно щоб у кожній зоні були сусіди з усіх сторін, було використано аналог техніки Padding, що часто використовується для аналізу зображень в згорткових нейронних мережах [7].

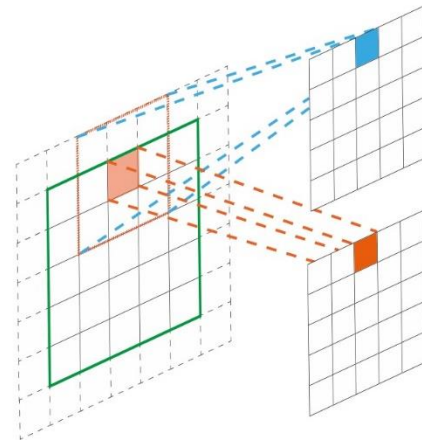


Рис. 3. Згортка карти для зон впливу (верхній, позначено синім кольором) та ситуації в поточній зоні (нижній, позначено оранжевим кольором)

Після аналізу дані про оточуючі зони та поточний стан у конкретній зоні передається в нейронну мережу, яка в свою чергу надає на вихід дані про напрям руху (або утримання поточної зон) і режим руху. На рис.4 надано схематичне зображення структури обробки даних для однієї зони після проведення перших двох етапів аналізу.

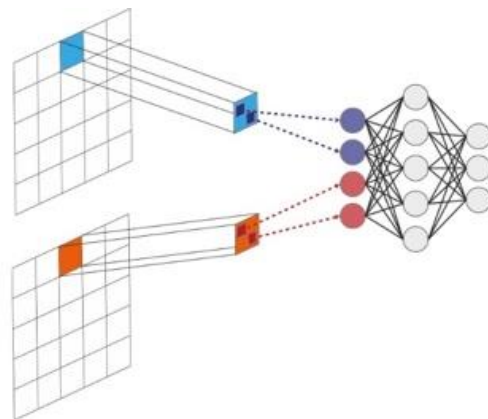


Рис. 4. Мережа для об'єднання результатів попередніх аналізів

Результатом для зони 3 є напрям, що можна виразити вектором $[1;-1]$ і режимом переходу 0.

На рис.5 кінцевий наказ, наданий групі в зоні 3. Відступ був необхідний через велику концентрацію одиниць супротивника, хоча в зоні перевага була за військами агента.

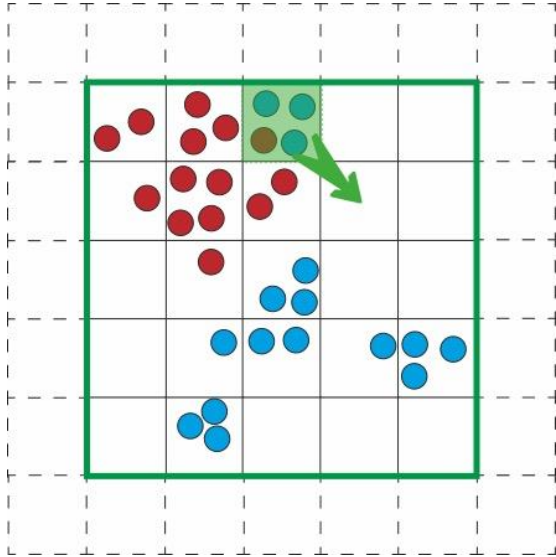


Рис. 5. Наказ групі в зоні 3

Результати

Проблемним було налаштування навчання системи, адже на відміну від типових задач, для яких використовують нейронні мережі, однозначно «правильного» варіанту не можна надати системі. Однак, система функцій, яка надає можливість отримувати дані про поточний стан військ, дозволяє моделювати ситуації з різними варіантами і надавати системі «правильний» варіант з розрахунку максимальної кількості балів. Це ускладнює навчання моделі з точки зору навантаження, однак дозволяє робити правильні висновки. Також, додано можливість коригувати дії за допомогою вказування правильного результату.

Основною метрикою, з допомогою якої вимірювалась ефективність системи, була оцінка ефективності використання бойових одиниць. Під час зіткнення бойових одиниць розраховувалась загальна оцінка групи військ, що вступає в територію контакту. Після зміни території зітк-

нення – переходу фокусу на іншу точку карти або знищення всіх бойових одиниць однієї з сторін – підраховується загальна оцінка. Загалом ця оцінка є різницею втрат гравців в поточній зоні, якщо втрати менші – дії в зоні були успішними. Ефективне використання військ в разі використання таких метрик можливо врахувати навіть при повній втраті бойових одиниць в межах території, оскільки вони можуть нанести пошкодження ворожим бойовим одиницям більше, ніж мають. Ефективність дій також можна визначити величиною оцінки – чим вона вища – тим вища різниця між втратами гравця та супротивника, а значить одиниці було використано максимально ефективно.

Для тестування використано декілька типових, попередньо заданих алгоритмів, що часто використовуються в стратегіях. Це види ботів з умовними позначенням «атакуючий», «захисний» та «збалансований».

Далі в табл.1 наводиться порівняльна характеристика результатів поєдинків бота під керуванням нейронної мережі та одного з ботів, які мають визначений алгоритм.

Для визначення ефективності навчання створено 3 варіанти моделі. Перший навчався за результатами тестувальників («правильний» варіант відповіді нейронної мережі задавався людиною). Другий варіант навчався виключно з використанням симуляції та підрахунку оцінки. Третій є комбінацією навчальних даних, які були використані.

Таблиця 1. Результати тестів

	атака	захист	баланс
1	67%	61%	64%
2	72%	65%	71%
3	75%	66%	73%

При аналізі результатів можна побачити, що ефективність нейронної мережі в порівнянні з стандартними ботами навіть на невеликих об'ємах тестових даних досить висока. Проти агресивного агента ефективність вища, оскільки в бою набагато простіше отримати перевагу – бот не чекає підкріплень і втрачає одиниці невеликими

групами. Нижчі результати з захисним боєм пов'язані з частим відступом останнього за межі території зіткнення, через необхідність отримання більшої маси військ або відсутності стратегічної точки для оборони. Відступ з зони не змінює цінності бойових одиниць, тому перемоги немає.

Загалом для оцінки проведено по 10 матчів. В кожному з матчів кількість боїв була різною. Цікавими результатами також є залежність кількості боїв в матчі від навченості нейронної мережі (рис.6). Як можна побачити з рис.6, на початку нейронна мережа заповнена випадковими коефіцієнтами, результат відповідний – швидка поразка через помилки управління. Зі збільшенням навченості нейронної мережі кількість зіткнень – активацій тактичного штучного інтелекту – за один матч зростала, оскільки до моменту закінчення бою проходило більше часу через вищу ефективність. Друга частина графіку показує ту частину навчання мережі, коли штучний інтелект став потужнішим за вбудований інтелект. Кількість зіткнень за матч почала падати, оскільки через високу ефективність використання зіткнення вели до великих втрат бойових одиниць, на відновлення яких потрібен час.

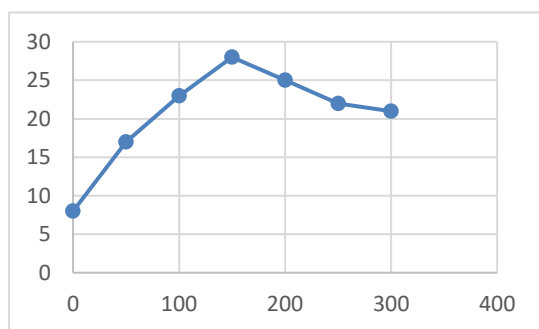


Рис. 6. Залежність кількості зіткнень від навчання нейронної мережі

Аналізуючи табл.1 можна дійти висновку, що комбінація навчання за результатами симуляції та навчання за прикладами створеними людиною, є оптимальним варіантом. Проблема навчання тільки за симуляцією в тому, що приклади від лю-

дини мають дещо більш узагальнене бачення картини і прораховування кроків на довгий час.

Ефективність штучного інтелекту можна збільшити навчанням мережі на більшій кількості даних, а також покращити алгоритм симуляції для повної автоматизації процесу.

Висновки

В ході дослідження розглянуто наукові та комерційні підходи для створення штучного інтелекту з використанням нейронних мереж. Розроблено тестове середовище, яке є спрощеною стратегією в реальному часі з обмеженою кількістю типів військ та дещо спрощеними ігровими правилами.

Для роботи в тестовому середовищі створено штучний інтелект для прийняття тактичних рішень для управління групами військ у зоні бою з супротивниками. В моделі роботи штучного інтелекту активно використовувались підходи, аналогічні тим, що використовуються в згорткових нейронних мережах.

Підсумовуючи дослідження можна з упевненістю сказати, що такий підхід є потенційно перспективним, основними напрямками для розвитку можуть стати використання різнотипних юнітів та урахування різної швидкості руху/дальності стрільби. Також перспективним напрямком є розвиток більш низькорівневого управління конкретними групами військ для збільшення їх ефективності.

Література

1. Stanescu M., Barriga, N. A., Hess A., Michael B. Evaluating Real-Time Strategy Game States Using Convolutional Neural Networks. – 2016. – 7 p.
2. Huang J., Yang W. A multi-size convolution neural network for RTS games winner prediction. *MATEC Web of Conferences*. – 2018. – 5 p.
3. Zhu J., Villareale J., Javvaji N., Risi S., Löwe M., Weigelt R., Harteveld C. Player-AI Interaction: What Neural Network Games Reveal About AI as Play. – 2021.
4. Synnaeve G., Bessière P. A Bayesian Model for RTS Units Control

Applied to StarCraft. *Proceedings of the 2011 IEEE Conference on Computational Intelligence and Games*. – 2011. – 7 p.

5. Sailer F., Buro M., Lanctot M. Adversarial Planning Through Strategy Simulation. *Proceedings of the IEEE Conference on Computational Intelligence and Games*. – Piscataway, NJ: Institute for Electrical and Electronics Engineers, 2007. – P. 80-87.

6. Szczepanski T., Aamodt A. Case-Based Reasoning for Improved

Micromanagement in Real-Time Strategy Games. *Case-Based Reasoning for Computer Games at the 8th International Conference on Case-Based Reasoning*, Seattle, WA, USA, 20-23 July, 2009. – 11 p.

7. Wiranata A., Wibowo S.A., Patmasari R., Rahmania R., Mayasari R. Investigation of Padding Schemes for Faster R-CNN on Vehicle Detection. – 2018. – P. 208-212.

Дорогий Я.Ю., Свириденко О.А.

ТАКТИЧНИЙ ШТУЧНИЙ ІНТЕЛЕКТ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СТРАТЕГІЇ В РЕАЛЬНОМУ ЧАСІ

Ця стаття містить розгляд та аналіз реалізації штучного інтелекту для гри в жанрі стратегії в реальному часу з використанням нейронної мережі. Штучний інтелект не є повноцінним агентом для стратегічного, тактичного та розвідано-аналітичного аспектів гри. Реалізація концентрується на вирішенні питань управління невеликими групами юнітів в ситуації контакту з супротивником та синхронізація дій груп з ціллю максимізації ефективності кожної окремої бойової одиниці.

Ключові слова: штучний інтелект, стратегії в реальному часі, відеоігри, нейронні мережі.

Dorohyy Ya.Yu., Svyrydenko O.A.

TACTICAL ARTIFICIAL INTELLIGENCE USING NEURAL NETWORK FOR REAL-TIME STRATEGY

This article discusses and analyzes the implementation of artificial intelligence for real-time strategy games using a neural network. Artificial intelligence is not a full-fledged agent for the strategic, tactical and intelligence-analytical aspects of the game. Implementation focuses on addressing the issues of managing small groups of units in a situation of contact with the enemy and synchronizing the actions of groups in order to maximize the effectiveness of each individual combat unit.

Keywords: artificial intelligence, real-time strategies, video games, neural networks.