УДК 004.896

Дробаха Д.А.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Корнага Я.І.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

УПРАВЛІННЯ 3D-МОДЕЛЛЮ АВТОМОБІЛЯ НА ТРЕКУ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

У статті висвітлюються практичні аспекти використання нейронних мереж для управління транспортними засобами на треку, зокрема як інструмента для збору інформації, а на основі отриманих даних – навчання автомобіля. Ми розглянемо, як можна навчити модель глибокого навчання, щоб передбачити кути повороту керма та допомогти віртуальному автомобілю управляти самим собою в симуляторі. У результаті отримуємо автоматичне прискорення й повороти керма автомобіля.

Ключові слова: *нейрон, нейронна мережа, архітектура, алгоритм навчання, керування процесами.*

Постановка проблеми. В останні кілька років ми спостерігаємо вибух інтересу до нейронних мереж, які успішно застосовуються в найрізноманітніших галузях (бізнесі, медицині, техніці, геології, фізиці). Нейронні мережі увійшли в практику всюди, де потрібно вирішувати завдання прогнозування, класифікації чи управління. Такий вражаючий успіх визначається декількома причинами:

– багато можливостей;

– простота використання.

Нейронні мережі привабливі з інтуїтивного погляду, бо вони засновані на примітивній біологічній моделі нервових систем. У майбутньому розвиток таких нейробіологічних моделей може привести до створення дійсно мислячих комп'ютерів.

Використовуючи нейронну мережу, ми хочемо, щоб транспортний засіб управляв собою сам, уникаючи перешкод. Ми досягаємо цього шляхом вибору відповідних входів/виходів і ретельного навчання нейронної мережі. Ми подаємо мережі відстані до найближчих перешкод навколо автомобіля, імітуючи зір водія-людини.

Аналіз останніх досліджень і публікацій. На цей проект нас фактично надихнула програма "End To End Learning For Self Driving Cars", створена дослідниками NVIDIA, яким вдалося розробити автомобіль для автономного водіння, навчивши нейронну мережу передбачати кути керма на основі даних кута повороту керма й зображень, захоплених трьома камерами (зліва, у центрі, праворуч), установленими перед автомобілем. Навчена модель здатна точно управляти автомобілем, використовуючи тільки центральну камеру.

Постановка завдання. У рамках цього проекту нам надається симулятор, написаний на Unity, який поставляється у двох режимах:

– режим тренування: ми вручну керуємо автомобілем і збираємо дані;

– автономний режим: автомобіль сам їздить на основі моделі, навченої за зібраними даними.

Журнал даних зберігається у файлі csv і містить шлях до зображень, а також кут керма, дросельної заслінки й швидкість. Нас цікавить тільки кут повороту керма й зображення для цього проекту.

Виклад основного матеріалу дослідження. Нейронні мережі з'явилися під час вивчення будови мозку. Наш мозок складається з 1011 клітин-нейронів, які посилають електричні сигнали один до одного. Кожен нейрон складається з одного або двох аксонів, які «видають результат», і великої кількості дендритів, які приймають вхідні електричні сигнали. Нейрону потрібна певна сила вхідного сигналу, який складається з усіх дендритів, щоб бути активованим. Після активації нейрон посилає електричний сигнал вниз по його аксону до інших нейронів. Зв'язки (аксонів і дендритів) зміцнюються, якщо вони часто використовуються.

Цей принцип застосовується в нейронних мережах менших масштабів. Сучасні комп'ютери не володіють потужностями обчислень, які створюють двадцять мільярдів нейронів, але навіть

із кількома нейронами нейронна мережа може давати розумну відповідь.

Нейрони організовуються в шари, як показано на малюнку 1. Вхідний шар буде мати входи, а залежно від міцності з'єднання з кожним нейроном у наступному шарі вхідний сигнал подається на наступний рівень. Міцність з'єднання називається вагою. Значення кожного нейрона в кожному шарі буде залежати від ваги зв'язку й значення нейронів попереднього шару.

Водія можна порівняти з функцією. Є безліч входів (те, що бачить водій). Ці дані обробляються мозком як функцією, і реакцією водія є вихід із функції.

Функція f(x) = y перетворює значення x (вимір) на y (вимір).

Ми використовуємо нейронну мережу зворотного поширення для «мозку» водія, оскільки такі нейронні мережі здатні апроксимувати будь-яку функцію з областями визначення й значення, які можуть мати кілька вимірів: F(x1,x2,..., xn) = y1, y2, ..., yn.

Це саме те, що нам потрібно, оскільки ми повинні працювати з кількома входами й виходами.

Коли нейронна мережа складається всього з декількох нейронів, ми можемо обчислити ваги, необхідні для отримання прийнятного результату. Але в міру збільшення кількості нейронів збільшується й складність обчислень. Мережу зворотного поширення можна навчити, що встановить необхідні ваги. Ми просто повинні надати шукані результати з відповідними входами.

Після навчання нейронна мережа буде реагувати й видавати результат, близький до бажаного під час подачі відомого результату, «відгадувати» правильну відповідь під час кожного входу, що не відповідає тому, який навчає.

Фактичні обчислення виходять за рамки статті. Є багато хороших книг, що пояснюють роботу мереж зі зворотним поширенням помилки.

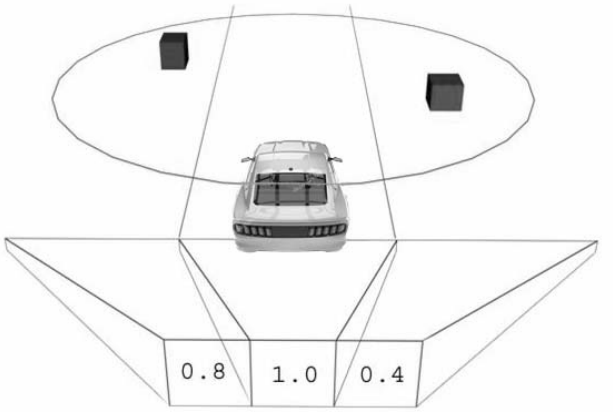
Нейронна мережа, яка використовується в нашому випадку, має 4 шари (рис. 2). Ми пробували різні комбінації (від трьох до шести шарів). Усе відмінно працювало з трьома шарами, але коли ми навчали мережу на наборі з двадцяти двох входів-виходів, наближення функції виявлялося недостатньо точним. П'ять і шість шарів чудово виконували своє завдання, але на навчання довелося витратити значний час (від 20 до 30 хвилин на PII), і коли ми запускали програму, потрібно було багато процесорного часу на обчислення.

У нашій мережі три нейрони у вхідному шарі й два – у виводі. Пізніше ми пояснимо, чому. Між ними два шари по вісім нейронів у кожному. Ми тестували шар із більшою й меншою кількістю нейронів і зупинилися на восьми, оскільки ця кількість дає прийнятний результат.

Під час вибору кількості нейронів необхідно мати на увазі, що кожен шар і кожен нейрон, доданий у систему, будуть збільшувати час, необхідний для розрахунку ваг.

Вхід. Яка інформація важлива для керування транспортним засобом? По-перше, ми повинні знати положення перешкоди відносно нас. Це положення праворуч, ліворуч від нас чи перед нами? Якщо є будівлі по обидва боки дороги, але нічого немає попереду, ми прискорюємося. Але якщо автомобіль зупинився перед нами, ми гальмуємо. По-друге, ми повинні знати відстань від нашої позиції до об'єкта. Якщо об'єкт розташований далеко, ми будемо продовжувати рух, поки він не наблизиться, і в цьому разі ми сповільнюємося чи зупиняємося.

Це саме та інформація, яку ми будемо використовувати для нашої нейронної мережі. Для простоти введемо три відносних напрямки (зліва, спереду й праворуч), а також відстані від перешкоди до транспортного засобу.



Визначимо поле зору нашого II-водія й складемо список об'єктів, які він бачить. Для простоти в нашому прикладі ми використовуємо коло, але могли б використовувати реальний усічений шістьма площинами конус. Тепер для кожного об'єкта в цьому колі перевіряємо, знаходиться він у лівому полі зору, у правому чи по центру.

На вхід у нейронну мережу подається масив float Vision [3]. Відстані до найближчої перешкоди зліва, у центрі й праворуч від транспортного засобу будуть зберігатися у Vision (0), Vision (1) і Vision (2) відповідно. На малюнку 3 показано, який вигляд має цей масив. Перешкода є зліва на відстані 80% від максимальної відстані, праворуч – на відстані 40%, немає ніяких перешкод по центру.

Таблиця 1

Вхідні нейрони Відстань до перешкоди			Вихідні нейрони	
Ліворуч	Центр	Праворуч	Прискорення	Напрямок
Немає перешкод	Немає перешкод	Немає перешкод	Повний газ	Прямо
Половина шляху	Немає перешкод	Немає перешкод	Невелике прискорення	Трішки направо
Немає перешкод	Немає перешкод	Половина шляху	Невелике прискорення	Трішки лівіше
Немає перешкод	Половина шляху	Немає перешкод	Гальмування	Трішки лівіше
Половина шляху	Немає перешкод	Половина шляху	Прискорення	Прямо
Зіткнення об'єкта	Зіткнення об'єкта	Зіткнення об'єкта	Зворотний рух	Вліво
Половина шляху	Половина шляху	Половина шляху	Без змін	Трішки лівіше
Зіткнення об'єкта	Немає перешкод	Немає перешкод	Гальмування	Повністю вправо
Немає перешкод	Немає перешкод	Зіткнення об'єкта	Гальмування	Повністю вліво
Немає перешкод	Зіткнення об'єкта	Немає перешкод	Зворотний хід	Вліво
Зіткнення об'єкта	Немає перешкод	Зіткнення об'єкта	Повний газ	Прямо
Зіткнення об'єкта	Зіткнення об'єкта	Немає перешкод	Зворотний хід	Повністю вправо
Немає перешкод	Зіткнення об'єкта	Зіткнення об'єкта	Зворотний хід	Повністю вліво
Об'єкт близько	Об'єкт близько	Об'єкт дуже близько	Без змін	Вліво
Об'єкт дуже близько	Об'єкт близько	Об'єкт близько	Без змін	Вправо
Зіткнення об'єкта	Об'єкт дуже близько	Об'єкт дуже близько	Гальмування	Повністю вправо
Об'єкт дуже близько	Об'єкт дуже близько	Зіткнення об'єкта	Гальмування	Повністю вліво
Зіткнення об'єкта	Об'єкт закритий	Об'єкт далеко	Без змін	Вправо
Об'єкт далеко	Об'єкт близько	Зіткнення об'єкта	Без змін	Вліво
Об'єкт дуже близько	Об'єкт близько	Ближче, ніж на половині шляху	Без змін	Повністю вправо
Об'єкт ближче, ніж на половині шляху	Об'єкт близько	Об'єкт дуже близько	Гальмування	Повністю вліво

Для того, щоб обчислити це, нам потрібна позиція (x, y) кожного об'єкта, положення (x, y) автомобіля й кут транспортного засобу. Нам також необхідні r (радіус кола) і dright, dleft – вектори між автомобілем і лініями Lright і Lleft. Ці лінії паралельні напрямку руху автомобіля. Обидва вектори – перпендикулярні лінії.

Хоча це 3D світ, уся математика двовимірна, оскільки автомобіль не може рухатися в третьому вимірі, тому що він не літає. Усі рівняння включають тільки x і y, але не z.

Насамперед ми обчислимо рівняння ліній Lright і Lleft, які допоможуть нам визначити, знаходиться перешкода праворуч, ліворуч або по центру від транспортного засобу.

Вихід. На виході ми повинні отримати інструкції щодо зміни швидкості автомобіля й напрямків. Це може бути прискорення, гальмування й кут повороту керма. Тому нам потрібні два виходи; один буде значенням прискорення/гальмування (гальмування – це просто негативне прискорення), а інший буде вказувати на зміну напряму.

Результат лежить між 0,0 і 1,0 з тієї ж причини, що й вхідні дані. Для прискорення 0,0 означає «повне гальмування»; 1,0 – «повний газ», а 0,5 – «відсутність гальмування чи прискорення». Для кермування 0,0 означає «повністю вліво», 1,0 – «повністю вправо», а 0,5 – «не змінювати напрямок». Таким чином, ми повинні перевести результати в значення, які ми можемо використовувати.

Слід зазначити, що «негативне прискорення» означає гальмування, якщо транспортний засіб рухається вперед, але це також означає «рухатися у зворотному напрямку», якщо автомобіль знаходиться в стані спокою. Крім того, «позитивне прискорення» означає гальмування, якщо транспортний засіб рухається у зворотному напрямку.

Навчання. Як згадувалося раніше, ми спочатку повинні навчити нейронну мережу. Нам необхідно створити набір входів і відповідних їм виходів.

Вибір правильних входів-виходів для навчання нейронної мережі, імовірно, найскладніша частина роботи. Нам довелося навчати мережу з безліччю даних, дивитися, як автомобіль діяв у навколишньому середовищі, а потім змінювати записи по мірі необхідності. Залежно від того, як ми навчаємо мережу, транспортний засіб може «коливатися» в деяких ситуаціях і здаватися нерухомим.

Складемо таблицю (табл. 1) різного положення перешкод щодо транспортного засобу та бажаної реакції ІІ.

І ось тепер можна перевести це в цифри в таблиці 2.

Таблиця 2

Вхідні нейрони			Вихідні нейрони	
Ліво-руч	По центру	Пра-воруч	Приско-рення	Напря-мок
1,0	1,0	1,0	1,0	0,5
0,5	1,0	1,0	0,6	0,7
1,0	1,0	0,5	0,6	0,3
1,0	0,5	1,0	0,3	0,4
0,5	1,0	0,5	0,7	0,5
0,0	0,0	0,0	0,2	0,2
0,5	0,5	0,5	0,5	0,4
0,0	1,0	1,0	0,4	0,9
1,0	1,0	0,0	0,4	0,1
1,0	0,0	1,0	0,2	0,2
0,0	1,0	0,0	1,0	0,5
0,0	0,0	1,0	0,3	0,8
1,0	0,0	0,0	0,3	0,2
0,3	0,4	0,1	0,5	0,3
0,1	0,4	0,3	0,5	0,7
0,0	0,1	0,2	0,3	0,9
0,2	0,1	0,0	0,3	0,1
0,0	0,3	0,6	0,5	0,8
0,6	0,3	0,0	0,5	0,2
0,2	0,3	0,4	0,5	0,9
0,4	0,3	0,2	0,4	0,1

Вхід:

0,0 – об'єкт майже стосується транспортного засобу;

1,0 – об'єкт на максимальній відстані від автомобіля або немає об'єкта в полі зору.

Вихід:

прискорення

0,0 – максимальне негативне прискорення (гальмування чи навпаки);

1,0 – максимальне позитивне прискорення.

Напрямок:

0,0 – повний поворот вліво;

0,5 – прямо;

1,0 – повний поворот вправо.

Висновки. Як ми щойно побачили, цей метод може бути поліпшений і застосований у найрізноманітніших галузях. Навіть якщо він не використовується для якоїсь корисної мети, нам все одно буде цікаво спостерігати, як система штучного інтелекту поводиться в навколишньому середовищі. Якщо спостерігати досить довго, ми зрозуміємо, що в складних умовах транспортний засіб не завжди буде йти одним і тим самим шляхом унаслідок невеликої різниці в рішенні у зв'язку з характером нейронної мережі. Автомобіль буде іноді їздити ліворуч від будівлі, а іноді – праворуч від неї.

Список літератури:

1. Rogers J. Object-Oriented Neural Network in C++, Academic Press, San Diego, CA, 1997.
2. Hagan M., Demuth H. and Beale M. Neural Network Design, PWS Publishing, Boston, MA, 1995.
3. Adams T. (2017). Training an artificial neural network.
4. Крисилов В., Олешко Д., Трутнев А.. Применение нейронных сетей в задачах интеллектуального анализа информации. Труды Одесского политехнического университета. Вып. 2 (8). 1999. С. 134.
5. Нейронные сети. STATISTICA Neural Networks. М.: Горячая линия-Телеком, 2000 г. С. 182.
6. Джеффри Е. Хинтон. Как обучаются нейронные сети. В мире науки. 1992. № 11–12. С. 103–107.
7. Садовой А., Сотник С. Алгоритмы обучения нейронных сетей будущего. URL: http://www.alicetele.com/~sergei/articles/algo/al go.htm.
8. Rumelhart B., Minton G., Williams R. Learning representations by back propagating error. Wature, 1986. V. 323. P. 1016–1028.
9. URL: http://www.accurate-automtion.com/Technology/Loflyte/loflyte.html.
10. URL: http://www.designation-systems.net/dusrm/app4/loflyte.html.

УПРАВЛЕНИЕ 3D-МОДЕЛЬЮ АВТОМОБИЛЯ НА ТРЕКЕ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

В статье освещаются практические аспекты использования нейронных сетей для управления транспортными средствами на треке, в частности как инструмента для сбора информации, а на основе полученных данных – обучения автомобиля. Мы рассмотрим, как можно научить модель глубокого обучения, чтобы предсказать углы поворота рулевого колеса и помочь виртуальному автомобилю управлять самим собой в симуляторе. В результате получаем автоматическое ускорение и повороты руля автомобиля.

Ключевые слова: *нейрон, нейронная сеть, архитектура, алгоритм обучения, управление процессами.*

3D-MODEL OF THE CAR ON THE TRACK USING A NEURAL NETWORK

The article highlights the practical aspects of the use of neural networks to control vehicles on the track, in particular as a tool for collecting information and on the basis of the obtained data training car. We will consider how to teach the model of deep learning to predict the rotation angles of the steering wheel and help virtual car driving itself in the simulator. As a result, we get automatic acceleration and steering turns.

Key words: *neuron, neural network, architecture, learning algorithm, process control.*