
Застосування Grasshopper Optimization Algorithm при навчанні нейронних мереж

Андрій Ляшкевич¹, Анастасія Дейнеко², Юлія Шевчук¹

*¹ Кафедра системного аналізу та теорії прийняття рішень
Київський національний університет імені Тараса Шевченка*

Київ, Україна

² IT STEP Університет

Львів, Україна

Вступ

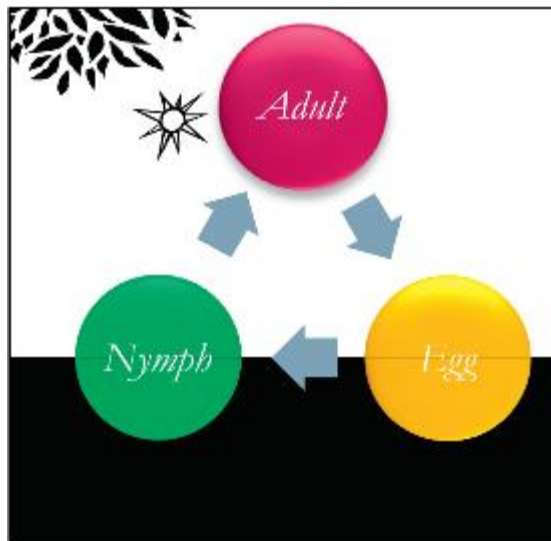
Процес знаходження найкращих значень змінних певної задачі для мінімізації або максимізації цільової функції називається оптимізацією. Оптимізаційні задачі існують у різних галузях знань. Щоб розв'язати оптимізаційну задачу, потрібно зробити кілька кроків. По-перше, слід визначити параметри задачі. Залежно від природи параметрів задачі можна класифікувати на неперервні та дискретні. По-друге, необхідно розпізнати обмеження, які накладаються на параметри. Обмеження поділяють оптимізаційні задачі на обмежені та необмежені. По-третє, необхідно дослідити та врахувати цілі даної задачі. У цьому випадку оптимізаційні задачі класифікуються на одноцільові та багатоцільові. Нарешті, на основі визначених типів параметрів, обмежень та кількості цілей слід обрати відповідний оптимізатор, який буде використовуватися для розв'язання задачі.

Вступ

Математична оптимізація в основному спирається на градієнтне формування відповідних функцій для знаходження оптимального рішення. Хоча такі методи все ще використовуються різними дослідниками, вони мають певні недоліки. Підходи математичної оптимізації страждають від захоплення локальних оптимумів. Це означає, що алгоритм припускає, що локальний розв'язок є глобальним розв'язком, і таким чином не може отримати глобальний оптимум.

Стохастичні методи покладаються на випадкові оператори, які дозволяють їм уникати локальних оптимумів. Всі вони починають процес оптимізації зі створення одного або набору випадкових розв'язків задачі. На відміну від математичних методів оптимізації, їм не потрібно обчислювати градієнт розв'язку, вони просто оцінюють розв'язки за допомогою цільової функції (функцій). Рішення про те, як покращити розв'язки, приймаються на основі розрахованих цільових значень. Таким чином, проблема розглядається як чорний ящик, що є дуже корисним механізмом при розв'язанні проблем з невідомими просторами пошуку.

Grasshopper Optimization Algorithm (GOA)



Стрибунці - це комахи. Вони вважаються шкідниками через шкоди, яку вони завдають рослинництву та сільському господарству. Життєвий цикл стрибунців показано на рис. 1. Хоча стрибунців у природі зазвичай бачать поодиночки, вони об'єднуються в один з найбільших роїв серед усіх істот. Розмір рою може мати континентальні масштаби і бути жахливим для масштабу континенту і є справжнім кошмаром для фермерів. Унікальність рою стрибунців полягає в тому, що роїова поведінка притаманна як у німф, так і в імаго. Мільйони стрибунців-німф стрибають і рухаються, як циліндри, що котяться. На своєму шляху вони поїдають майже майже всю рослинність. Після такої поведінки, коли вони стають дорослими, вони утворюють рій у повітрі. Так стрибунці мігрують на великі відстані.

Основною характеристикою рою на стадії личинки є повільний рух і маленькі кроки стрибунців. На противагу цьому, великі відстані та різкі рухи на великі відстані є основною рисою рою у дорослому віці. Ще однією важливою характеристикою рою є пошук джерел їжі.

Grasshopper Optimization Algorithm (GOA)

Математична модель, що використовується для моделювання ройової поведінки стрибунців виглядає наступним чином:

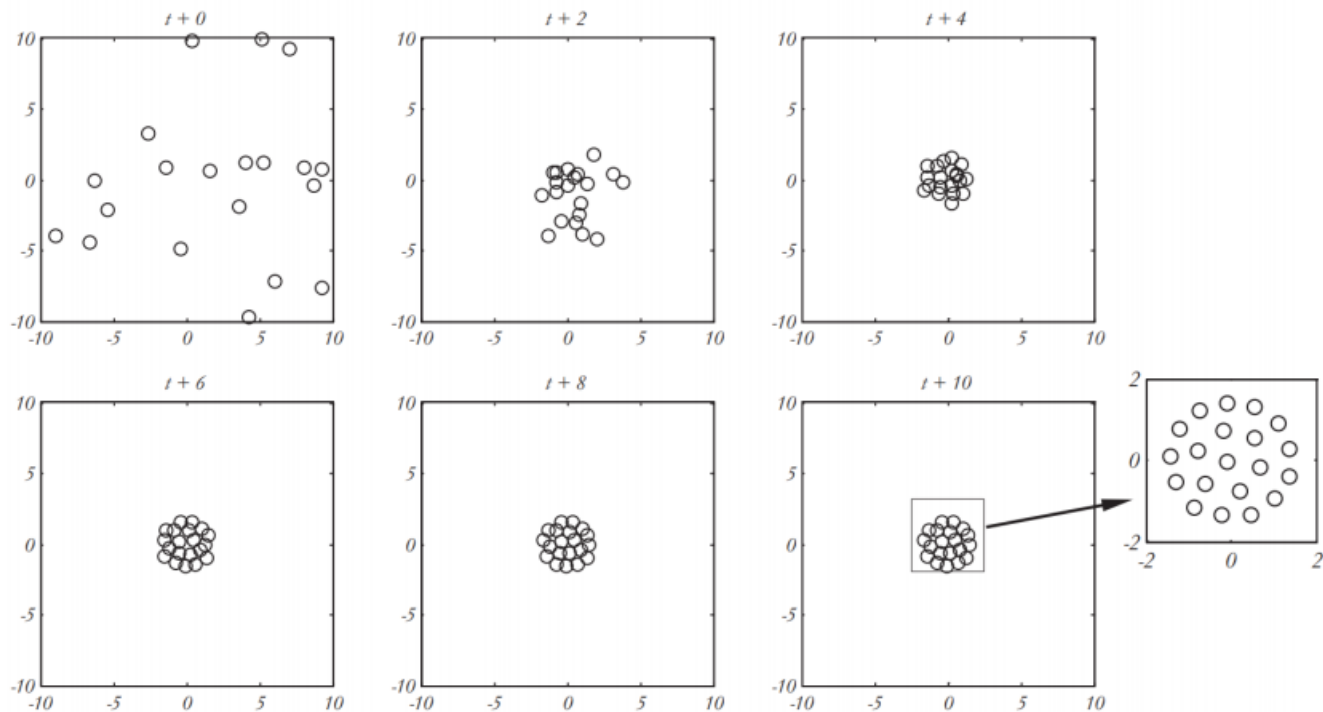
$$X_i = S_i + G_i + A_i,$$

де X_i визначає положення i -го стрибунця, S_i – соціальна взаємодія, G_i - сила гравітації на i -го стрибунця, і A_i показує адвекцію вітру.

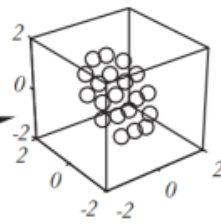
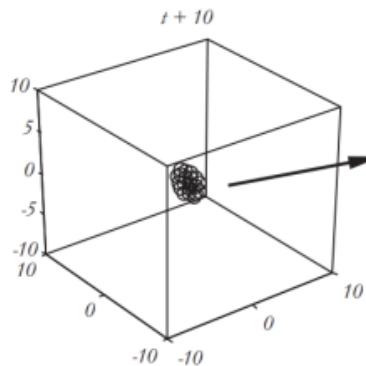
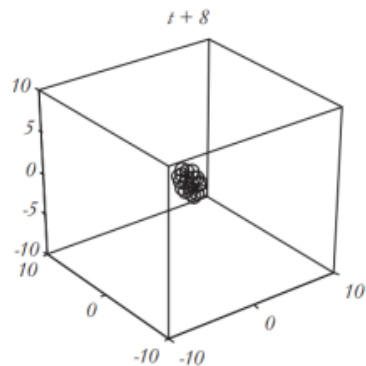
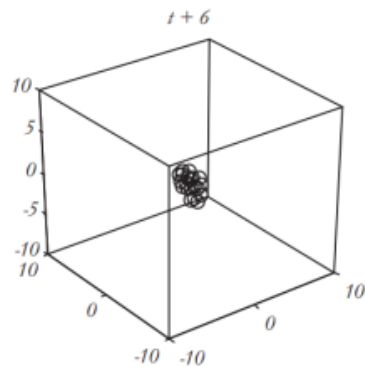
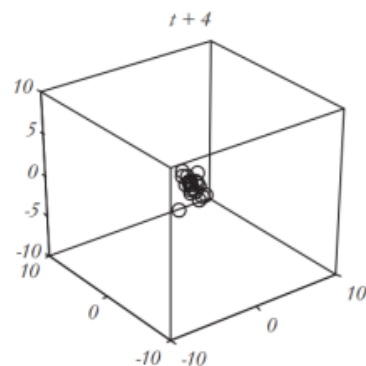
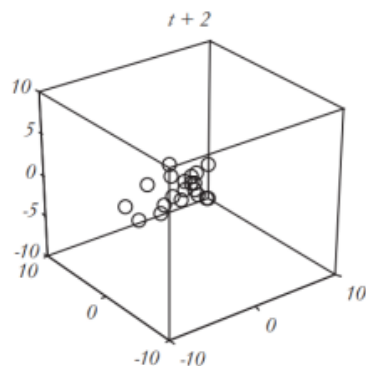
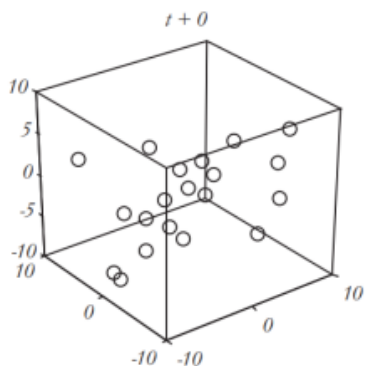
Потрібно зазначити, що дана модель враховує розташування поточного стрибунця відносно інших стрибунців. Фактично, ми розглянули стан усіх стрибунців, щоб визначити місцеперебування пошукових агентів навколо цілі.

GOA оновлює позицію пошукового агента на основі його поточної на основі його поточної позиції, глобального найкращого результату та позицій всіх інших пошукових агентів.

Поведінка рою в двовимірному просторі



Поведінка рою в тривимірному просторі



Алгоритм навчання нейронної мережі Forward-Forward

Алгоритм Forward-Forward – це жадібна багат шарова процедура навчання, натхненна машинами Больцмана та Noise Contrastive Estimation. Ідея полягає в тому, щоб замінити прямий і зворотний проходи backpropagation двома прямими проходами вперед, які працюють точно так само, як і один, але на різних даних і з протилежними цілями. Позитивний прохід працює з реальними даними і коригує ваги, щоб підвищити якість кожного прихованого шару. Негативний прохід оперує з "негативними даними" і коригує ваги для зменшення якості в кожному прихованому шарі. У цій статті досліджуються дві різні міри якості – сума квадратів нейронних активностей та від'ємна сума квадратів активностей, але існує багато інших мір.

Припустимо, що функція корисності для шару є просто сумою квадратів активностей випрямлених лінійних нейронів у цьому шарі. Мета навчання полягає у тому, щоб зробити функцію доброякісності значно вищою за деяке порогове значення для реальних даних і значно нижчою за це значення для від'ємних даних. Точніше, мета полягає в тому, щоб правильно класифікувати вхідні вектори як позитивні або негативні дані, коли ймовірність того, що вхідний вектор є позитивним (тобто реальним), задається шляхом застосування логістичної функції σ до якості, мінус деякий поріг θ :

$$p(\text{positive}) = \sigma\left(\sum_j y_j^2 - \theta\right),$$

де y_j - активність прихованої одиниці j до нормалізації шару. Негативні дані можуть бути передбачені нейронною мережею за допомогою низхідних зв'язків або отримані ззовні.

Навчання декількох шарів представлення за допомогою простої пошарової функції доброякісності

FFNN-оптимізації Grasshopper Optimization Algorithm

В контексті FFNN-оптимізації Grasshopper Optimisation Algorithm застосовано наступним чином:

1. Ініціалізація: Випадковим чином ініціалізуються ваги та зміщення FFNN в межах заданого діапазону. Генерується початкова популяція стрибунців, кожен з яких представляє потенційний розв'язок.
2. Оцінка: Оцінюється придатність кожного стрибунця шляхом навчання FFNN з використанням поточного набору ваг та зміщень. Пристосованість зазвичай визначається продуктивністю мережі на перевіркому наборі даних, де менша помилка або вища точність вказує на кращу пристосованість.
3. Рух стрибунця: GOA використовує різні стратегії руху для дослідження простору пошуку. Ці стратегії включають пошук ділянок з їжею (тобто локальний пошук), слідування за найкращими кониками (тобто глобальний пошук) та стрибки на нові позиції.
4. Локальний пошук: Випадковий вибір деяких стрибунців для локального пошуку шляхом зміни їхніх позицій у межах невеликого району. Це допомагає використовувати локальні області пошукового простору і вдосконалювати рішення.
5. Глобальний пошук: Визначити найкращого стрибунця з найвищим fitness значенням та оновити позиції інших стрибунців, щоб наблизитися до позиції найкращого стрибунця. Це заохочує дослідження перспективних областей у просторі пошуку.
6. Стрибки: Випадковий вибір деяких стрибунців для виконання стрибків шляхом випадкового оновлення їхніх позицій у просторі пошуку. Це дозволяє алгоритму уникати локальних оптимумів і досліджувати нові регіони.
7. Оновити FFNN: Після переміщення стрибунців оновлюємо ваги та зсуви FFNN на основі позицій стрибунців. Цей крок передбачає коригування параметрів за допомогою відповідних математичних операцій, таких як кросовер і мутація, для покращення загальної пристосованості мережі.
8. Завершення роботи: Повторюємо кроки 2-7 до тих пір, поки не буде виконано умову завершення, наприклад, досягнення максимальної кількості ітерацій або бажаного рівня пристосованості. В кінці процесу оптимізації FFNN отримає оптимізовані ваги та зміщення, які можна використовувати для точних прогнозів на невидимих даних.

Підсумки

У даній роботі представлено нейронну мережу типу Forward-Forward із використанням Grasshopper Optimization Algorithm як оптимізатора. Дана нейронна мережа сприяє ефективному розв'язанню задач які ставляться перед нейронними мережами, а також розвитку нейронних мереж та оптимізаторів розглянутих типів та може бути корисною для науковців, аналітиків та інших фахівців, що працюють у даній галузі. Для подальшого розвитку даної роботи у майбутньому, можна застосувати принципи snapshot unsambly та cycle learning rate, що повинно збільшити ефективність нейронної мережі.

Серед ключових напрямів подальшого розвитку моделі нейронної мережі Forward-Forward, яка використовує алгоритм оптимізації Grasshopper Optimization Algorithm (GOA) - розширення архітектури мережі, оптимізація параметрів GOA, застосування моделі до реальних даних, порівняння з іншими алгоритмами, розробка додаткових функцій GOA, використання ансамблевих моделей та адаптація до різних завдань. Виконання цих досліджень має потенціал покращити ефективність та універсальність розробленої моделі, сприяючи розвитку області машинного навчання.

Дякую за увагу