

Нечітка Інкрементна Самоорганізована Багатошарова Мапа та її Послідовне Навчання

Євгенія Іванова
кафедра штучного інтелекту
Харківський національний університет
радіоелектроніки
Харків, Україна
yevheniia.ivanova@nure.ua

Євгеній Бодянський
кафедра штучного інтелекту
Харківський національний університет
радіоелектроніки
Харків, Україна
yevgeniy.bodyanskiy@nure.ua

Fuzzy Incremental Self-Organizing Multilayer Map and its Sequential Learning

Yevheniia Ivanova
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yevheniia.ivanova@nure.ua

Yevgeniy Bodyanskiy
Artificial Intelligence Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yevgeniy.bodyanskiy@nure.ua

Анотація—В статті запропоновано нечітку інкрементну самоорганізовану багатошарову мапу для вирішення задачі послідовного кластерування. Архітектура самоорганізованої мапи складається з нечіткого шару виводу, що дозволяє проводити нечітке кластерування в послідовному режимі.

Abstract—The paper proposes a fuzzy incremental self-organizing multilayer map for solving the problem of serial data streams clustering. The architecture of self-organizing map consists of a fuzzy output layer that allows for fuzzy clustering in sequential mode.

Ключові слова—нечітке кластерування, інкрементна самоорганізована мапа, послідовне навчання, інтелектуальний аналіз даних, багатошарова нейронна мережа.

Keywords—fuzzy clustering, incremental self-organizing map, consistent learning, data science, multilayer neural network.

III. ВСТУП

В епоху інформаційних технологій динамічне зростання обсягу даних викликало стрімкий розвиток наукового напрямку штучний інтелект. Існує багато напрямків цієї науки і найбільш великими є машинне навчання та інтелектуальний аналіз даних.

В рамках цих наук вирішується доволі багато завдань, а саме таких як регресія, класифікація, кластерування, прогнозування та інші. Найбільш ефективні методи реалізовані на основі штучних нейронних мереж.

Рішення задачі кластерування масивів даних здійснюється в режимі навчання без вчителя. Вхідна множина об'єктів розбивається на групи зі схожими властивостями. Серед алгоритмів кластерування виділяється клас методів нечіткого кластерування. Застосування методів цього класу дозволяє формалізувати різного роду невизначеності, які завжди існують при вирішенні реальних завдань. Нечітке кластерування є синтезом ідей кластерного аналізу та теорії нечітких множин.

Самоорганізована інкрементна нейронна мережа використовується для навчання без вчителя в онлайн режимі. Ключове завдання полягає в розділенні немічених нестаціонарних вхідних даних в різні класи без попереднього знання о кількості існуючих класів і основних характеристиках вибірки даних. Також завдяки використанню запропонованого алгоритму є можливість вивчити топологію вхідних даних.



Для реалізації цих цілей потрібно визначити ключові аспекти, а саме концепція введення нових вузлів, адаптивна швидкість навчання та видалення вузлів з низькою щільністю. SOINN (self-organizing incremental neural network) [3] використовує двошарову конкурентну мережу, в якій алгоритм навчання для першого та другого шарів однаковий.

IV. НЕЧІТКА ІНКРЕМЕНТНА САМООРГАНІЗОВНА БАГАТОШАРОВА МАПА ТА ЇЇ ПОСЛІДОВНЕ НАВЧАННЯ

A. Самоорганізовна інкрементна нейронна мережа

SOINN складається з двох шарів. двошарову конкурентну мережу. Перший шар аналізує щільність розподілу вхідних даних і використовує вузли та ребра для представлення цього розподілу. Другий шар розділяє дані на кластери шляхом розташування області вхідних даних з низькою щільністю та використовує менше вузлів ніж перший шар. Коли навчання другого шару завершено, SOINN підлаштовує кількість кластерів і видає прототипи вузлів кожного кластера. Він також використовує аналогічний алгоритм навчання для першого та другого шарів.

Конкурентне навчання Хебба представляє парадигму для побудови зв'язків між вузлами нейронної мережі. Конкурентне правило Хебба можна описати таким чином: для кожного вхідного сигналу з'єднується два найближчі вузли, вимірювані на основі евклідової відстані. Граф є оптимальною топологією в режимі послідовного навчання та протягом усього процесу навчання мережі вузли, сусідні на ранній стадії, можуть стати не сусідніми на більш розвиненій стадії. Тим самим стає необхідним видаляти з'єднання, які були оновлені дуже давно.

Для завдання класифікації без учителя необхідно визначити, чи відноситься вхідний приклад до раніше вивчених кластерів або до нового кластеру. Припустимо, що два приклади належать до одного кластеру, якщо евклідова відстань між ними менша за порогову відстань T . Якщо T занадто велике, всі зразки будуть віднесені до одного кластеру. Якщо T занадто мале, кожен зразок буде утворювати ізольований одиничний кластер. Щоб отримати «природні» кластери, T має бути більшим ніж відстань всередині кластера і менша, ніж відстань між кластерами.

Для обох шарів обчислюється порогова відстань T . У першому шарі встановлюється вхідний сигнал як новий вузол (перший вузол нового кластеру), коли відстань між сигналом і найближчим вузлом (або другим найближчим вузлом) перевищує порогове значення T , яке постійно адаптується до поточної інформації. У другому шарі розраховується відстань всередині кожного кластера і відстані між кластерами на основі тих вузлів, що були згенеровані в першому шарі. В результаті отримується постійна порогова відстань T_c відповідно до відстаней всередині кластера та відстаней між кластерами.

Коли вхідний вектор подається на вхід SOINN [3], він знаходить найближчий вузол (переможець) і другий найближчий вузол (другий переможець) вхідного вектору. Потім використовуючи порогові критерії подібності

мережа визначає, чи належить вхідний вектор кластеру першого переможця або другого.

Перший рівень SOINN адаптивно оновлює поріг подібності кожного вузла, оскільки розподіл вхідних даних невідомий. Якщо вузол i -тий має сусідні вузли, поріг подібності T_i обчислюється з використанням максимальної відстані між вузлом i -тим і сусідніми його вузлами згідно з формулою (1), а якщо вузол i не має сусідніх вузлів за формулою (2):

$$T_i = \max_{j \in N_i} \|w_i - w_j\|, \quad (1)$$

$$T_i = \min_{j \in N \setminus \{i\}} \|w_i - w_j\|. \quad (2)$$

На рис. 1 представлена блок-схема алгоритму SOINN.

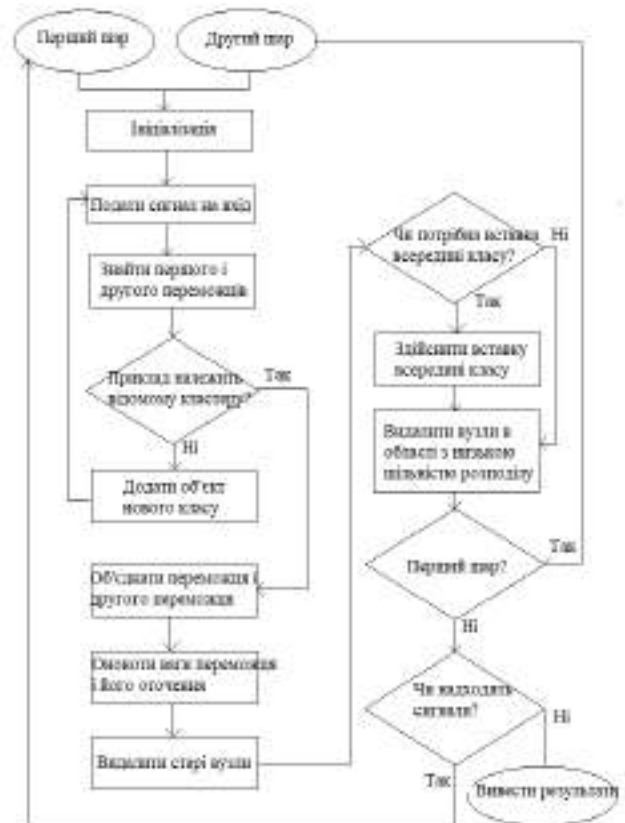


Рис. 1. Блок-схема алгоритму SOINN

Вхідний вектор визначається в мережі як новий вузол для представлення першого вузла нового класу, якщо відстань між вхідним вектором і переможцем або другим переможцем більше, ніж поріг схожості переможця або другого переможця. Якщо вхідний вектор визначений як належний до одного кластеру як у переможця або другого переможця і якщо немає ребра, що з'єднує переможця і другого переможця. Тоді з'єднуємо переможця і другого переможця за допомогою ребра і встановлюємо вік ребра рівним нулю. Надалі збільшуємо вік всіх ребр, пов'язаних з переможцем, на одиницю.



Після цього оновлюється вектор ваги переможця та його сусідні вузли за формулами (3) і (4). Використаємо i для позначення вузла переможця і M_i , щоб показати час, коли він став переможцем. Зміна ваги переможця Δw_i і зміна ваги сусіднього вузла Δw_j визначаються як:

$$\Delta w_i = \frac{1}{M_i} (w_s - w_i), \quad (3)$$

$$\Delta w_j = \frac{1}{100M_i} (w_s - w_j). \quad (4)$$

Якщо вік одного ребра більше заданого параметра $age_{\max}$, то це ребро видаляється.

Після λ ітерацій (λ є таймером) SOINN [2] вставляє нові вузли в положення, де накопичена помилка є великою. Скасовується вставка, якщо вона не може зменшити помилку. Вставка називається внутрікласовою вставкою, тому що новий вставлений вузол знаходиться в межах існуючого класу. Крім того, під час вставки новий клас не буде створено. Потім SOINN знаходить вузли, кількість сусідів яких менша або дорівнює одиниці і видаляє ці вузли.

В. Нечітка інкрементна самоорганізовна мапа

Відмінною особливістю нечіткого кластерування є той факт, що кожен об'єкт може відноситися до кожного кластеру з певним ступенем належності. Правильний вибір функції належності є важливим для коректної роботи алгоритму. Аналіз збіжності процесів конкурентного самонавчання, проведений М. Котреля і Дж. Фортом [1], показав, що в процесі настройки синаптичних ваг, повинен зменшуватися не тільки крок пошуку, але і параметр ширини функції сусідства. Чим більша відстань переможця від об'єкта, тим менше корегується його синаптична вага.

Функція сусідства повинна бути ширше на початку процесу навчання, і її ширина повинна зменшуватися з часом таким чином, щоб до кінця цього процесу здійснювалася підгонка тільки безпосередніх сусідів переможця і другого переможця.

В якості функції сусідства найчастіше використовують Гауссіан, параметри якого змінюються від часу, найчастіше лінійно. Ця функція має наступну формулу (5):

$$u(j, l, k) = \exp\left(-\frac{d^2(w_j^*(k), w_l(k))}{\sigma^2}\right), \quad (5)$$

в якій σ – параметр ширини функції належності; $w_j^*(k)$ – синаптична вага нейронна переможця, $w_l(k)$ – синаптичні ваги мережі.

Цей параметр необхідно налаштовувати і вводити формулу для його розрахунку. Тому в атестаційній роботі запропоновано функцію належності, параметри якої налаштовувалися без введення нових.

Використання функцій сусідства призводить до модифікованого правила навчання інкрементної самоорганізовної мапи і реалізує принцип «переможець отримує більше». Для спостереження $x(k)$, яке поступає на вхід, визначається перший і другий вузол переможець $w_j^*(k)$ і наближається до спостереження за формулою (6):

$$w_j^*(k+1) = w_j^*(k) + \eta(k)u^2(j, l, k)(x(k) - w_j^*(k)). \quad (6)$$

Всі інші вузли (для спрощення обчислень це може бути фіксована кількість вузлів або всі вузли ближче деякої порогової відстані) перераховуються за формулою (7):

$$w_l(k+1) = w_l(k) + \eta(k)u^2(j, l, k)(x(k) - w_l(k)). \quad (7)$$

Змінна $u(j, l, k)$ – функція належності. На прикладі алгоритму FCM (fuzzy c-means) визначається функція сусідства і розраховується за формулою (8):

$$u_j(k) = \frac{d^2(x(k), c_j)}{\sum_{j=1}^m d^2(x(k), c_j)} = \frac{\|x(k) - c_j\|^2}{\sum_{j=1}^m \|x(k) - c_j\|^2}. \quad (8)$$

Після розрахунку функції належності, перевизначаються координати центроїдів кожного кластера згідно формули (9):

$$c_j = \frac{\sum_{k=1}^N u_j^2(k)x(k)}{\sum_{k=1}^N u_j^2(k)}. \quad (9)$$

Формула (9) – це математичне трактування функції сусідства Кошіан. Перевагою запропонованої функції належності є те, що на відміну від Гауссіана значення функції на нескінченності не близько до нуля. Що сприяє тому, що в навчанні беруть участь всі вузли.

В процесі роботи нечітка інкрементна самоорганізовна мапа реагує на отримання спостереження $x(k)$, наближаючи до нього вузол-переможець $w_j^*(k)$ відповідно до принципів конкурентного навчання. При цьому інші вузли $w_l(k)$ наближаються до вузла-переможця пропорційно ступеня сусідства між ними.

V. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

Для реалізації нечіткого методу послідовного кластерування на основі інкрементної самоорганізовної нейронної мережі була обрана середовище програмування PyCharm і мова Python.

Нижче описуються базові бібліотеки, які були використані в ході написання програми FSOINN. Ці бібліотеки роблять з мови програмування Python потужний і надійний інструмент для аналізу і візуалізації даних.

NumPy – основна бібліотека Python, яка спрощує роботу з векторами і матрицями. Містить готові методи для самих різних операцій: від створення, зміни форми,



множення і розрахунку детермінанта матриць до рішення лінійних рівнянь і сингулярного розкладання. В реалізації FSOINN використовувалась ця бібліотека для створення масивів, в яких зберігався поданий на вхід сигнал і в цьому виді передавався на обробку. Також застосовувалися функції знаходження суми, максимального і мінімального значення, наприклад, для формул (1) і (2).

SciPy – бібліотека ґрунтується на NumPy і розширює її можливості. Включає методи лінійної алгебри і методи для роботи з ймовірними розподілами, інтегральним обчисленням і перетвореннями Фур'є. Також містить підпрограми, такі як чисельна інтеграція і оптимізація, для підвищення продуктивності. Як і всі бібліотеки Python описана в документації. Використовується бібліотека `scipy.sparse.dok_matrix` – словник ключів на основі розрідженої матриці.

Бібліотека Matplotlib для створення двовимірних діаграм і графіків. З її допомогою можна побудувати будь-який графік. Однак для складної візуалізації потрібно більше коду, ніж в розвинених бібліотеках.

Для перевірки роботи розробленого алгоритму кластерування, були згенеровані вибірки різних форм. Ці вибірки були обрані, бо головна мета – показати характеристики різних алгоритмів кластерування на наборах даних, які є «цікавими», але все ще в 2D.

Процес обробки вхідних даних включає в себе такі етапи:

- користувач задає параметри системі;
- завантажує в систему дані;
- система задає початкові параметри мережі;
- система проводить нечітке кластерування;
- система підраховує функції належності;
- система розраховує час навчання нейронної мережі;
- дані відображаються в 2D (елементи фарбуються згідно з результатами кластерування).

Для експерименту взяті вибірки складної неопуклої форми, а саме так звані "бублик" і дві спіралі.

На рис. 2 показано дві вибірки і час навчання нейронної мережі. Причому в першій вибірці один клас знаходиться всередині іншого, а в другій вони лінійно неподільні. Але запропонований метод кластерного аналізу точно відокремлює одне кільце від іншого.

Вибірка складалася з 2000 прикладів і нейронна мережа навчилася за 28 секунд.

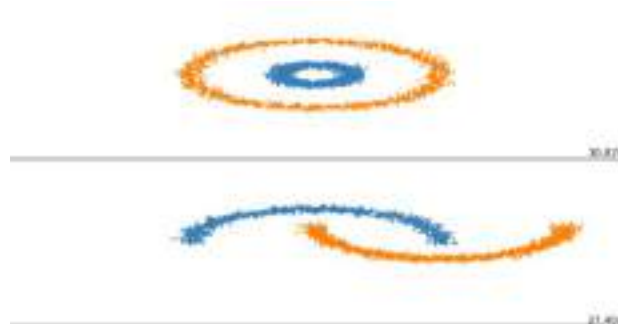


Рис. 2. Візуалізація кластерування методом FSOINN

VI. ВИСНОВКИ

Запропоновано метод кластерного аналізу нечіткої інкрементної самоорганізованої мапи в режимі послідовного навчання на основі самоорганізованої інкрементної нейронної мережі. Запропонований алгоритм дозволяє без апіорного знання кількості кластерів вирішувати задачу нечіткого послідовного кластерування і візуалізувати топологічну структуру даних в двовірному просторі для ефективного аналізу даних.

Розроблений алгоритм ефективно працює з неопуклими наборами даних і досить простий в реалізації, але для проведення навчання необхідно налаштовувати параметри. Він відноситься до особливого класу штучних нейронних мереж, а саме до самоорганізованих мап, які вирішують завдання кластерування і топологічної структури даних.

В статті представлені результати, які відповідно до поставленої мети є рішенням актуального завдання нечіткого кластерування масивів даних в умовах довільної кількості кластерів. Теоретично була виведена, обґрунтована і запропонована функція належності, а також доведені її переваги.

ЛІТЕРАТУРА REFERENCES

- [6] Shen F. An Algorithm for Incremental Unsupervised Learning and Topology Representation / F. Shen, H. Osamu. 2006. P. 105.
- [7] . Shen F. An enhanced self-organizing incremental neural network for online unsupervised learning / F. Shen, H. Osamu. Neural Networks, 2007. №20. P. 893–903
- [8] Shen F. A fast nearest neighbor classifier based on self-organizing incremental neural network / F. Shen, H. Osamu. Neural Networks, 2008. №21. P. 1537–1547.
- [9] Shen F. An incremental network for on-line unsupervised classification and topology learning / F. Shen, H. Osamu. Neural Networks, 2006. №90. P. 106.
- [10] Borgelt C. Prototype-based Classification and Clustering [Text] / C. Borgelt. Magdeburg, 2005. P. 350.
- [11] Bodyanskiy Ye. Data streams fast EM-fuzzy clustering based on Kohonen's self-learning [Text] / Ye. Bodyanskiy, A. Deineko, Y. Kutsenko, O. Zayika // The 1th IEEE International Conference on Data Stream Mining & Processing (DSMP 2016): proc. of int. conf., Lviv, August 23-27, 2016. Lviv, 2016. P. 309-313.

