

4) створенням відповідних інтерфейсів та без зміни фізичного розташування об'єктів можна віднести окремих предмет до потрібного розділу, віртуальної колекції або декількох колекцій. Такий підхід схематично показано на рис. 5. Це дає змогу емулювати бази даних з врахуванням специфіки подання фондів та створювати враження роботи з окремою базою даних.

5. Висновки

Усе сказане про принципи створення інформаційного порталу засобами ZOPE дає змогу стверджувати, що проблеми єдності та роздільності структури бази даних музейної інформації органічно поєднуються, доповнюються та не суперечать одна одній.

Запропонований підхід до реалізації завдань, викладених у концепції інформатизації Львівського державного природознавчого музею [1], означає перехід установи на якісно вищий рівень її діяльності, створює передумови входження до телекомунікаційного середовища НАН України, національної комп'ютерної мережі освіти і науки UNREN, забезпечує інтеграцію з міжнародними телекомунікаційними мережами GEANT, створює можливість для належного подання музею у світовому інформаційному просторі.

1. Климшин О.С., Войчишин В.К. Наукова концепція інформатизації Державного природознавчого музею НАН України // *Наук. зап. Держ. природозн. музею.* – Львів, 2003. – Т. 18. – С. 5–16.
2. Юрєнева Т.Ю. *Музееведение: Учебник для высшей школы.* – М.: Академический Проект, 2003. – 560 с.
3. Morrison A. *Words and images // Museums journal.* – 1990. – Vol. 90, № 8. – P. 30-31.
4. Prochak M. *Multimedia is the message // Museums journal.* – 1990. – Vol. 90, № 8 – P. 25-27.
5. Спикльмайр С., Фридли К., Спикльмайр Дж., Брэнд К. *ZOPE. Разработка Web-приложений и управление контентом* – М.: ДМК Пресс, 2003.
6. Климшин О.С. *Наукова концепція фондової роботи Державного природознавчого музею НАН України // Наук. зап. Держ. природозн. музею.* – Львів, 2001. – Т. 16. – С. 3–32.

УДК 519.7

О. В. Годич

Львівський національний університет ім. І. Франка,
кафедра дискретного аналізу та інтелектуальних систем

КЛАСТЕРИЗАЦІЯ ДАНИХ НЕЙРОМЕРЕЖЕЮ ADD

© Годич О. В., 2005

Запропоновано нову топологію штучної нейромережі та метод її навчання під назвою ADD. Нейромережа ADD разом зі спеціально розробленим для неї методом навчання має подолати недоліки, притаманні для штучних нейромереж, які базуються на топології SOM. Особливість запропонованої топології нейромережі полягає у поданні чітко окресленої гіперкубічної області у вхідному просторі, дані з якої розпізнаються мета-нейроном.

In the paper that is presented, a new neural network topology is proposed and a new learning method which utilises the distinct features of the proposed neural network is considered. The main virtue of the proposed neural network is the ability to represent a hypercubic area in the input space which defines the boundaries of the data represented by the meta-neuron.

1. Вступ

Задачі кластеризації стосуються теорії видобування даних, яка має важливі практичні застосування [1, 2, 3]. Під кластеризацією розуміють задачу організації деякої множини об'єктів у значущі групи. Тут "значущість" є навмисно нечіткою характеристикою, яка залежить від

конкретного випадку застосування кластеризації. Так, наприклад, якщо евклідова відстань між векторами n -вимірного дійснозначного простору може бути використана як ознака подібності, то “значущість” груп таких векторів може визначатись їхньою близькістю в сенсі евклідової відстані.

Удосконалення існуючих та розробка нових методів кластеризації має важливе теоретичне та практичне значення. Топологія, покладена в основу штучної нейромережі SOM (Self-Organizing Maps), є найпоширенішою нейромережевою технологією, використовуваною для розв’язування задач кластеризації, і є фундацією цілої низки топологій, породжених від SOM. Класичну модель нейромережі SOM та метод її навчання було запропоновано Кохоненом [4].

Надалі під вхідними даними розумітимемо множину векторів $X \subset R^n$. Де R^n є n -вимірним дійснозначним простором, який називатимемо вхідним простором. Вектори $x \in X \subset R^n$ називатимемо вхідними векторами.

Виділяємо три основні недоліки нейромережі SOM Кохонена:

Статична, наперед визначена структура нейромережі. Одним із вагомих недоліків SOM Кохонена є вимога визначення структури нейромережі перед її навчанням. Невдалий підбір структури безпосередньо впливає на результат роботи навченої нейромережі. У загальному випадку структуру SOM підбирають способом спроб та помилок. Зміна у вхідних даних, потенційно, може призвести до вимоги із зміни структури SOM для забезпечення адекватної якості навчання.

SOM забезпечує “точкову” апроксимацію. Відповідно до означення нейрона найкращого наближення [5] (best matching unit, BMU) для довільного вектора із вхідного простору завжди існує BMU. Тобто, деяка група векторів із вхідного простору представляється одним ваговим вектором. Межі такої групи векторів можливо встановити лише накладанням деякого штучного обмеження (наприклад, куля із центром у ваговому векторі та емпірично встановленим радіусом). Такий підхід значною мірою перекреслює основну властивість SOM – здатність до самоорганізації. Водночас, існування BMU для довільного вектора x із вхідного простору не гарантує “доброї” апроксимації вхідних даних і може призвести до хибного розпізнавання кластерів. Така ситуація є актуальною, в першу чергу, коли навчену SOM використовують для кластеризації даних, які містять кластери, що не існували або були недостатньо представлені в навчальній множині.

Неможливість донавчання SOM. Донавчання SOM фактично зводиться до перенавчання нейромережі, що зумовлено особливостями навчального алгоритму Кохонена. Така ситуація унеможливорює використання вже набутого нейромережею досвіду під час попереднього навчання. Із збільшенням навчальної множини збільшується тривалість процесу навчання.

У теорії штучних нейромереж, які стосуються SOM, велика кількість робіт присвячена вирішенню першої проблеми – динамічній розбудові структури нейромереж для подолання двох основних причин похибок SOM [6]:

Невідповідність між розмірностями вхідного простору та структури SOM.

Невідповідність у представленні структури вхідних даних у випадку існування в них декількох незв’язних множин.

Так, зокрема, у праці [7] авторами запропоновано метод GCS (Growing Cell Structure), який динамічно додає та віднімає нейрони в нейронній ґратці, керуючись деяким евристичним критерієм, що ґрунтується на частоті успіху нейрона (кількість разів, коли нейрон ставав BMU) чи його сумарній похибці (відхилення від вхідних даних, для яких нейрон є BMU).

У праці [8] запропоновано метод IGG (Incremental Grid Growing), який ґрунтується на SOM Кохонена в навчанні, але динамічно нарощує нейрони. Додавання та віднімання нейронів у цьому випадку базується на сумарній похибці нейронів на межі двовимірної нейронної ґратки. Такий підхід вирішує проблему представлення даних з незв’язними множинами, але успадковує проблему представлення даних у випадку відмінності в розмірності між вхідними даними та нейронною ґраткою.

Автори праці [9] розробили метод GSOM (Growing Self-Organizing Map), який за своєю ідеєю нагадує IGG, але нарощування нейронів відбувається в довільному напрямку. Такий підхід забезпечує вирішення проблеми невідповідності між розмірностями вхідних даних та ґратки. Основним недоліком цього методу є складність обчислень.

Автори висвітлюють теоретичні засади нової топології нейромережі ADD та методу її навчання, а також порівнюють застосування нейромережі ADD та SOM на двох наборах даних різної складності [6].

2. Модель мета-нейрона. Метод навчання add. Функція add

Мета-нейрон (базова модель). Модель штучного нейрона, яку використовують у SOM, являє собою дві алгебричні величини: координати нейрона в просторі нейромережі (у моделі Кохонена це двовимірний простір) та ваговий вектор w , який має розмірність, що дорівнює розмірності вхідного простору, дані з якого аналізують на виявлення кластерів. Координати нейронів у SOM є статичними й наперед визначеними, що зумовлює потребу у визначенні структури нейромережі перед її навчанням. Вагові вектори нейронів зазвичай [5] ініціалізуються рівномірно розподіленими випадковими значеннями із проміжку $[0, 1]$ або випадково вибраними значеннями з навчальної множини даних. Недоліки таких підходів для навчання SOM методом Кохонена та потенційні шляхи їхнього подолання обговорюються в [10].

Впродовж процесу навчання SOM вагові вектори w_j нейронів коригуються так, що кожен із них відповідає за представлення певної підмножини векторів із навчальної множини вхідних даних. Графічну інтерпретацію зв'язку між вхідними даними та ваговим вектором w_j нейрона i наведено на рис. 1. На цьому рисунку схематично зображено множину двовимірних даних у вигляді області, обведеної кривою; підмножину даних, для яких нейроном i є BMU, подано заштрихованою областю; символ “ \star ” зображає ваговий вектор w_j у вхідному просторі; хвиляста лінія символізує розмежування простору вхідних даних та SOM; символ “ \ominus ” зображає деякий вектор із вхідної множини даних, який представляється нейроном i . Ваговий вектор w_j трактується як проєкція нейрона i у просторі вхідних даних, а сам нейрон i є проєкцією множини векторів з вхідного простору, для яких він є BMU [5] на SOM.

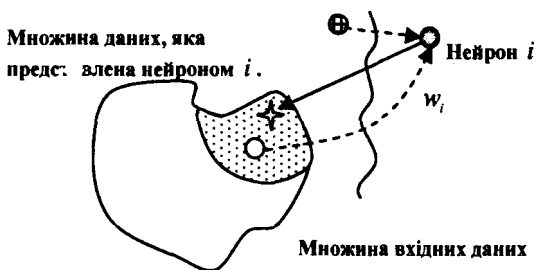


Рис. 1. Зв'язок між вхідними даними та ваговим вектором w_j нейрона-переможця i у SOM.

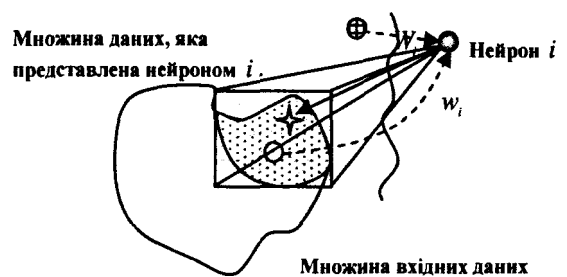


Рис. 2. Зв'язок між вхідними даними та ваговим вектором W_i нейрона-переможця i у ADD.

Недоліком такої моделі є той факт, що якщо в процесі пошуку кластерів за допомогою навчальної SOM дані міститимуть кластери, яких не було у навчальній множині, то результат кластеризації буде некоректним. Підґрунтям такого твердження є те, що для кожного вектора із вхідного простору завжди існує нейрон найкращого наближення, який визначається співвідношенням (1).

$$i(x) = \arg \min_j \|x - w_j\|, \quad (1)$$

де x – вектор із вхідного простору, w_j – ваговий вектор нейрона $j = \overline{1, N}$, N – кількість нейронів у SOM.

Для ілюстрації хибного розпізнавання на рис. 1 символом “ \oplus ” зображено вектор, який не належить навчальній множині. Нехай відповідно до формули (1) нейрон i є його BMU. За таких умов вектор “ \oplus ” буде помилково трактуватись як частини кластера, за представлення якого відповідає нейрон i .

Для подолання описаного недоліку запропоновано модифікувати модель штучного нейрона введенням інтервального вагового вектора W_i , де кожна компонента є не точковим значенням, а інтервалом. Позначатимемо ваговий вектор як $w_i = (w_i^1, w_i^2)$, а інтервальный ваговий вектор як $W_i = (W_i^1, W_i^2) = ([\underline{w}_i^1, \overline{w}_i^1], [\underline{w}_i^2, \overline{w}_i^2])$. Введення інтервального вагового вектора забезпечує чітке визначення меж за кожним виміром вхідного простору, в яких нейрон вважатиметься ефективним. Інтервальный ваговий вектор W_i не замінює точковий ваговий вектор w_j , а відіграє виключно роль обмеження. Спосіб побудови W_i є основою запропонованого навчального методу, який обговорюється нижче.

Надалі вживатимемо термін “традиційний”, або “точковий” нейрон, коли мова йде про модель нейрона з точковим ваговим вектором, та “мета-нейрон”, коли мова йде про інтервальну модель нейрона. Зазначимо, що для спрощення подання матеріалу статті використовуватимемо двовимірний дійснозначний простір R^2 , якщо не вказано інакше.

На рис. 2 схематично зображено запропоновану модель мета-нейрона та її зв'язок із вхідними даними. Більшість елементів на рис. 2 мають те саме навантаження, що й на рис. 1. Ключовою відмінністю є прямокутна область навколо виділеної підмножини вхідних даних, вектори якої представлені нейроном i . Цей прямокутник зображає інтервальний ваговий вектор W_i мета-нейрона. У загальному випадку інтервальний вектор W_i задає гіперкубічну область, яка для двовимірного простору є прямокутником. Отже, говоритимемо про проекцію нейрона i на гіперкубічну область у вхідному просторі, а не лише на точкове значення, як це є для традиційної моделі нейрона в SOM.

Розглянемо вже обговорену для традиційної моделі нейрона проблему хибного розпізнавання векторів із вхідного простору. На рис. 2 вектор, який не належить до навчальної множини, представлено символом “ \oplus ”. Позначимо цей вектор як $x_0 = (x_0^1, x_0^2)$. Нехай нейрон i відповідно до формули (1) є ВМУ для значення $x = x_0$. Щоби перевірити, чи нейрон i може представити вхідний вектор x_0 , введемо умову (2):

$$x_0^l \in W_i^l, \quad (2)$$

де $l = \overline{1,2}$. Якщо умова (2) виконується для всіх компонент вхідного вектора x_0 , то вважатимемо нейрон i таким, що представляє цей вхідний вектор. Як зображено на рис. 2, жодна з компонент вектора x_0 не належить відповідній їй компоненті з інтервального вагового вектора W_i . Відтак, можна стверджувати, що вектор x_0 не належить проекції нейрона i на вхідний простір і не представлений цим нейроном. Отже, для моделі мета-нейрона умова (1) є необхідною, а умова (2) достатньою для того, щоб нейрон i був ВМУ при довільному вхідному векторі x .

Нейромережа та метод ADD, розширена модель мета-нейрона. Для подолання обмежень, які зумовлені статичною та визначеною наперед структурою нейромережі SOM, розроблено динамічну нейромережу ADD. Ця нейромережа у своїй основі базується на мета-нейронах як структурних елементах та методи навчання ADD. Окрім обчислення вагових векторів нейронів метод ADD контролює структуру нейромережі. Назва методу ADD (від англ. average dimensional distance – усередкована розмірнісна відстань) відображає спосіб коригування вагових векторів та динамічної модифікації структури нейромережі. Відтак, принципи функціонування методу ADD є невід'ємною частиною обговорення нейромережі ADD.

Основною концепцією, яку покладено в метод ADD, є інформаційна насиченість мета-нейронів. Під інформаційною насиченістю розумітимемо значення деякої дійснозначної функції $f(i)$, де i – індекс мета-нейрона у нейромережі. Чим вище значення інформаційної насиченості, тим вищою вважають адаптацію мета-нейрона для представлення вхідних значень. Тобто, модифікація вагових векторів та структури нейромережі методом ADD спрямована на збільшення значення інформаційної насиченості $f(i)$ для всіх $i = \overline{1, N_d}$, де N_d – кількість нейронів у нейромережі. Індекс d у N_d вказує на те, що кількість нейронів не фіксована й може змінюватись динамічно впродовж навчання.

Виділимо три етапи методу ADD:

- **Змагання.** Подаючи вхідний вектор x , для кожного мета-нейрона в нейромережі обчислюють деяку класифікуючу функцію. Мета-нейрон із найбільшим значенням класифікуючої функції оголошують переможцем.

- **Адаптація.** Етап адаптації спрямований на модифікацію мета-нейрона-переможця для підвищення його інформаційної насиченості.

- **Взаємодія.** Під взаємодією мета-нейронів розуміємо їхнє об'єднання з метою утворення мета-нейрона з вищим рівнем інформаційної насиченості.

Для обговорення цих трьох етапів введемо *розширену модель мета-нейрона та функцію ADD*. Одне з можливих трактувань числового інтервалу є інтервал як множина чисел між його початком та кінцем [11]. Виходячи з такого трактування, компоненти інтервального вагового вектора

W в моделі мета-нейрона можна представити як множини дійсних чисел. Для цього введемо вектор $S = (S^1, S^2)$, компоненти якого є множинами. При цьому інтервальний ваговий вектор W можна записати як $W = (W^1, W^2) = ([\min(S^1), \max(S^1)], [\min(S^2), \max(S^2)])$. Модель мета-нейрона, яка складається з вагового вектора w , інтервального вагового вектора W та вектора S , називатимемо *розширеною моделлю* мета-нейрона.

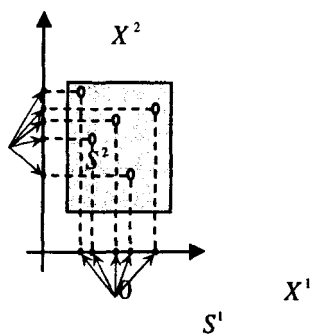


Рис. 3. Геометрична інтерпретація вектора S

На рис. 3 схематично представлено інтерпретацію вектора S , де задано п'ять точок $x_1 = (x_1^1, x_1^2)$, $x_2 = (x_2^1, x_2^2)$, $x_3 = (x_3^1, x_3^2)$, $x_4 = (x_4^1, x_4^2)$ та $x_5 = (x_5^1, x_5^2)$, зображених символом "o". Проекції цих точок на кожну з осей представлено символом ".". З урахуванням того, що всі вхідні дані, зображені на рис. 3, можуть бути представлені одним мета-нейроном, вектор S матиме вигляд $S = (\{x_1^1, x_2^1, x_3^1, x_4^1, x_5^1\}, \{x_1^2, x_2^2, x_3^2, x_4^2, x_5^2\})$. Отже, змістовне навантаження компонент вектора S полягає в представленні проекцій векторів із вхідного простору на осі координат, за відображення яких відповідає мета-нейрон. Додатково на компоненти вектора S накладемо умову, що кожна із множин може містити лише унікальні значення, тобто без повторень. Надалі модель деякого i -го мета-нейрона позначатимемо як трійку векторів (w_i, W_i, S_i) .

Інформаційну насиченість обчислюють за методом ADD за допомогою однойменної функції ADD. Для обговорення цієї функції введемо такі позначення: $|W^l|$ – ширина інтервалу W^l , $|S^l|$ – потужність, або кількість елементів у множині S^l , де $l = \overline{1, n}$, n – розмірність вхідного простору. Зазначимо, що використання n -вимірного простору в подальшому обговоренні необхідне для відображення загальності підходу.

Інформаційна насиченість мета-нейрона визначатиметься для кожного виміру $l = \overline{1, n}$ вхідного простору. В основу такого обчислення покладено компоненти інтервального вагового вектора $W = (W^1, \dots, W^n)$ та вектора множин $S = (S^1, \dots, S^n)$, а саме W^l та S^l . Функцію інформаційної насиченості ADD, розроблену емпіричним шляхом, представлено формулами (4) та (5).

$$f(W^l, S^l) = 1 - \exp \left(- \frac{\left(\sum_{s \in S^l} g(s, W^l, S^l) \right)^2}{2 \times |W^l|^2} \right) \quad (3)$$

$$g(\tilde{s}, W^l, S^l) = \exp \left(- \frac{\left(\sum_{s \in S^l} |s - \tilde{s}| \right)^2 \times \left(\frac{|W^l|}{|S^l|} \right)^2}{2 \times |W^l|} \right) \times \exp \left(- \frac{\min_{s \in S^l, s \neq \tilde{s}} |s - \tilde{s}|}{2 \times |W^l|} \right) \quad (4)$$

Функція $f(W^l, S^l)$ визначає співвідношення між квадратом суми значень функції $g(\tilde{s}, W^l, S^l)$ за всіма $s \in S^l$, та квадратом ширини інтервалу W^l . Перший множник функції $g(\tilde{s}, W^l, S^l)$ визначає співвідношення між сумою квадрата відстані переданого елемента $\tilde{s} \in S^l$ до решти $s \in S^l, s \neq \tilde{s}$, квадратом "оптимальної" відстані, та шириною інтервалу W^l . Під "оптимальною" відстанню розуміємо

вираз $\frac{|W^l|}{|S^l|}$, який обчислює відстань між елементами множини S^l на інтервалі W^l за умов їхнього

рівномірного розподілу. Метою методу ADD є таке перетворення множин S^l , щоб елементи $s \in S^l$ були розподілені максимально рівномірно на інтервалі W^l . Зазначимо, що перший множник у формулі (4) можна спростити, але представлена форма краще сприяє поясненню.

Другий множник функції $g(\tilde{s}, W^l, S^l)$ визначає відношення мінімальної відстані переданого елемента $\tilde{s} \in S^l$ до решти $s \in S^l, s \neq \tilde{s}$, та шириною інтервалу W^l . Головною мотивацією внесення цього множника у формулу (4) потрібно вважати вплив надто близьких елементів у множині S^l відносно відстані між рештою елементів.

Оскільки експонента в співвідношенні (3) набуває свого значення на проміжку $[0, 1]$, то функція $f(W^l, S^l)$ також набуває своїх значень на цьому проміжку. Причому, чим вище значення $f(W^l, S^l)$, тим вищою є інформаційна насиченість множини S^l на інтервалі W^l . Функцію $f(W^l, S^l)$ називатимемо функцією ADD мета-нейрона для виміру $l = \overline{1, n}$. На рис. 4 зображено графіки зміни значень функцією ADD за двома вимірами для деякого мета-нейрона впродовж навчання методом ADD. Вісь OY відповідає значенням функції, а вісь OX – порядковим номерам ітерацій процесу навчання, коли відбувалась модифікація мета-нейрона (на проміжних ітераціях саме цей мета-нейрон не модифікувався). Метод ADD керується функцією ADD для модифікації мета-нейронів з метою підвищення їхньої інформаційної насиченості. Деталі процесу модифікації мета-нейронів представлено в обговоренні етапу адаптації. Перейдемо до детальнішого обговорення етапів методу ADD.



Рис. 4. Графіки зміни функції ADD за двома вимірами для одного із мета-нейронів впродовж навчального процесу

Етап змагання. Етап змагання між мета-нейронами за представлення деякого вхідного вектора $x \in R^n$ задається класифікувальною функцією. Результатом змагання є мета-нейрон-переможець, який представляє вхідний вектор x найкраще в сенсі класифікувальної функції. Для методу ADD, так само як і для методу Кохонена [10], як класифікувальну функцію вибрано евклідову відстань між вхідним вектором x та ваговим вектором w мета-нейрона. Отже, визначення мета-нейрона-переможця зводиться до обчислення співвідношення (1). А саме, $i(x) = \arg \min_j \|x - w_j\|$, де x –

вхідний вектор, w_j – ваговий вектор мета-нейрона $j = \overline{1, N}$, а N – кількість мета-нейронів у нейромережі на поточній ітерації методу ADD. Як буде висвітлено під час обговорення етапу адаптації, кількість мета-нейронів у нейромережі змінюється динамічно. Відтак, число N не є постійним, як це є для SOM Кохонена.

Зазначимо, що мета-нейрон, який був визначений переможцем за результатом змагання, не обов'язково є ВМУ, оскільки за означенням виконання умови (1) є лише необхідною умовою.

Етап адаптації. Під адаптацією в методі ADD розуміємо деяку модифікацію інтервального вагового вектора W та вектора S мета-нейрона-переможця етапу змагання, яка призводить до підвищення інформаційної насиченості мета-нейрона продовж навчального процесу. Зауважимо, що підвищення інформаційної насиченості мета-нейрона під час кожної його модифікації не вимагається, тобто можливі коливання. Прикладом цього є рис. 4.

Нехай $x_* = (x_*^1, \dots, x_*^n)$ є деякий вхідний вектор із навчальної множини $X \subset R^n$. І нехай мета-нейрон i з моделлю $(w_i = (w_i^1, \dots, w_i^n), W_i = (W_i^1, \dots, W_i^n), S_i = (S_i^1, \dots, S_i^n))$ є мета-нейроном-переможцем, визначеним для вхідного вектора x_* на етапі змагання. Етап адаптації передбачає два види модифікацій:

1. Розширення мета-нейрона. Розширення полягає у внесенні компонент вхідного вектора x_*^l , $l = \overline{1, n}$ до відповідних компонент-множин S_i^l . А саме:

$$S_i^l \leftarrow \{x_*^l\} + S_i^l, l = \overline{1, n} \quad (5)$$

Після обробки компонент вхідного вектора за формулою (5) визначаємо інтервальний ваговий вектор W_i відповідно до співвідношення (6) та ваговий вектор w_i за формулою (7).

$$W_i^l = [\min(S_i^l), \max(S_i^l)], \quad (6)$$

$$w_i^l = \frac{\sum_{s \in S_i^l} s}{|S_i^l|}, \quad (7)$$

де $l = \overline{1, n}$, n – розмірність вхідного простору.

Отже, розширення мета-нейрона-переможця i при деякому поданому вхідному векторі x_* визначається співвідношеннями (5), (6) та (7). Причому, співвідношення (5) має бути виконаним перед (6) та (7). Надалі позначатимемо розширення мета-нейрона i як i^{ex} (від англ. extended). Відповідно, модель мета-нейрона i^{ex} можна записати як $(w_{i^{ex}}, W_{i^{ex}}, S_{i^{ex}})$.

2. Розщеплення мета-нейрона. Вхідний вектор x_* може порушити стан мета-нейрона i в тому сенсі, що розширення мета-нейрона i може призвести до зниження його інформаційної насиченості. Тобто, значення функції ADD для розширеного мета-нейрона i^{ex} набуватиме менших значень, ніж для мета-нейрона i . Процес розщеплення передбачає породження нового мета-нейрона i^{new} , для якого вектори $(w_{i^{new}} = (0, \dots, 0), W_{i^{new}} = ([0, 0], \dots, [0, 0]), S_{i^{new}} = (\emptyset, \dots, \emptyset))$ утворюються в результаті розщеплення мета-нейрона i^{ex} .

Тобто, якщо умова (8) виконується хоча б для одного значення $l = \overline{1, n}$, то мета-нейрон i^{ex} повинен бути розщеплений (від англ. split).

$$f(W_{i^{ex}}^l, S_{i^{ex}}^l) < f(W_i^l, S_i^l), l = \overline{1, n} \quad (8)$$

Розщеплення мета-нейрона i^{ex} полягає в поділі множин $S_{i^{ex}}^l$, $l = \overline{1, n}$, для яких виконується умова (8). Розглядатимемо два випадки:

1) значення $x_*^l \in W_i^l$. У цьому випадку множину $S_{i,ex}^l$ розбиваємо на дві підмножини: $S_{i,ex}^l{}' = \{s \in S_{i,ex}^l \mid s < x_*^l\}$ та $S_{i,ex}^l{}'' = \{s \in S_{i,ex}^l \mid s > x_*^l\}$. При цьому, значення x_*^l додається до тієї підмножини, до якої воно є ближчим у сенсі евклідової норми. Тобто, якщо виконується умова $|x_*^l - \max(S_{i,ex}^l{}')| < |x_*^l - \min(S_{i,ex}^l{}'')|$, то $S_{i,ex}^l \leftarrow \{x_*^l\} + S_{i,ex}^l{}'$. Інакше $S_{i,ex}^l \leftarrow \{x_*^l\} + S_{i,ex}^l{}''$.

Обчислюємо $S_{i,ex}^l$ та $S_{i,new}^l$ відповідно до формул (9) та (10):

$$S_{i,ex}^l \leftarrow S_{i,ex}^l{}' \quad (9)$$

$$S_{i,new}^l \leftarrow S_{i,ex}^l{}'' \quad (10)$$

2) значення $x_*^l \notin W_i^l$. У цьому випадку потрібно повернути множину $S_{i,ex}^l$ до попереднього стану:

$$S_{i,ex}^l \leftarrow S_i^l \quad (11)$$

А значення x_*^l вибираємо за множину $S_{i,new}^l$ для породженого нейрона i^{new} :

$$S_{i,new}^l \leftarrow \{x_*^l\} \quad (12)$$

Після завершення формування множин $S_{i,ex}^l$ та $S_{i,new}^l$ необхідно визначити компоненти інтервальних вагових векторів $W_{i,ex}^l$ та $W_{i,new}^l$ за формулою (6), а також компоненти вагових векторів $w_{i,ex}^l$ та $w_{i,new}^l$ за формулою (7). Важливо зазначити, що для тих $l = \overline{1, n}$, для яких умова (8) не виконується, застосовуємо випадок (2).

Звернемо увагу на те, що розщеплення мета-нейронів веде до динамічної розбудови нейромережі природним, у сенсі інформаційної насиченості, способом. Динаміка зміни кількості мета-нейронів у нейромережі від початку та до завершення навчального процесу зображено на рис. 13–14. Ці графіки відповідають навчальному процесу з використанням тестових множин, які обговорювалися в підрозділі “Приклади застосування нейромережі ADD. Порівняння з SOM.”. Також зазначимо, що запропонований підхід адаптації, зокрема, визначення вагового вектора w_i за формулою (3), модифікує мета-нейрони так, що вони відображають топологічно близькі вхідні вектори, тим самим імітуючи структуру вхідних даних.

Етапи змагання та адаптації чергуються один за одним впродовж навчального процесу, тобто при поданні кожного вхідного вектора відбувається змагання, а результуючий мета-нейрон-переможець адаптується. Фаза змагання/адаптації в методі ADD називається *фазою впорядкування*. Основною метою цієї фази є адаптація та розбудова нейромережі для представлення вхідних даних. Кількість ітерацій фази впорядкування збігається з кількістю вхідних векторів у навчальній множині. Це зумовлено тим, що змагання та адаптацію проводять для кожного вектора з навчальної множини лише один раз (див. також розділ “Алгоритм ADD”).

Етап взаємодії. Етап взаємодії відповідає *фазі консолідації* нейромережі ADD. Цей етап реалізує об’єднання мета-нейронів за умови, що значення інформаційної насиченості є вищим для об’єданого мета-нейрона, ніж їхні індивідуальні значення до об’єднання. Використання саме інформаційної насиченості, тобто функції ADD як критерію об’єднання, узгоджується з етапом адаптації, де функцію ADD використовують для породження нових мета-нейронів. Додатково накладаємо умову, що лише мета-нейрон та його найближчий сусід можуть об’єднуватись, а сусідство визначається в сенсі топології вхідного простору. Тобто деякий мета-нейрон i може бути об’єднаний лише зі своїм найближчим сусідом i^{bmu} , який знаходиться відповідно до співвідношення (13), яке є модифікацією співвідношення (1). А саме, вхідним значенням є ваговий вектор w_i

та накладається умова $j \neq i$ для уникнення вибору мета-нейрона i як свого найближчого сусіда. Зауважимо, що схожу концепцію сусідства використано в методі “нейронного газу” [12].

$$i^{bmu}(w_i) = \arg \min_{j, j \neq i} \|w_i - w_j\| \quad (13)$$

Процедура об'єднання двох мета-нейронів i та i^{bmu} полягає в об'єднанні їхніх векторів S_i та $S_{i^{bmu}}$. Формула (14) реалізує покомпонентне об'єднання множин S_i та $S_{i^{bmu}}$. Об'єднаний мета-нейрон позначатимемо як \tilde{i} , вектори $W_{\tilde{i}}$ та $w_{\tilde{i}}$ якого обчислюються за формулами (6) і (7) відповідно.

$$S_{\tilde{i}}^l \leftarrow S_i^l \cup S_{i^{bmu}}^l, \quad l = \overline{1, n} \quad (14)$$

де n – розмірність вхідного простору.

Після об'єднання мета-нейрони i та i^{bmu} видаляють із нейромережі, а мета-нейрон \tilde{i} – вносять. Звернемо увагу, що означення множини $S_{\tilde{i}}^l$ не можуть містити повторів, тобто об'єднання множин за формулою (14) передбачає уникнення дублювань. Умова об'єднання визначається таким співвідношенням:

$$f(W_{\tilde{i}}^l, S_{\tilde{i}}^l) > f(W_i^l, S_i^l), \quad l = \overline{1, n} \quad (15)$$

Важливо зазначити, що об'єднання мета-нейронів i та i^{bmu} можливе лише тоді, якщо умова (15) виконується для всіх $l = \overline{1, n}$. Можливість об'єднання визначається для кожного мета-нейрона в нейромережі. Етап взаємодії вважаємо завершеним, якщо відповідно до умови (15) жоден мета-нейрон у нейромережі не може бути об'єднаним.

Алгоритм ADD

Нехай маємо навчальну множину $X \subset R^n$, потужність якої є $|X| = K$. Елементами множини X є вектори $x_k \in X$. Нейромережу розглядатимемо як список, елементи якого є пронумеровані. Кожен елемент цього списку є мета-нейроном, які можна видаляти зі списку або додавати в його кінець. Кількість мета-нейронів у нейромережі позначимо N . Індекс k представляє номер поточного вхідного вектора $x_k \in X$, $k = \overline{1, K}$. Індекс $i = \overline{1, N}$ відповідає за порядковий номер мета-нейронів у нейромережі.

Ініціалізація. Ініціалізуємо індекси $k \leftarrow 1$, $i \leftarrow 1$. Утворимо перший мета-нейрон (w_i, W_i, S_i) , де $w_i = (x_k^1, \dots, x_k^n)$, $W_i = ([x_k^1, x_k^1], \dots, [x_k^n, x_k^n])$ та $S_i = (\{x_k^1\}, \dots, \{x_k^n\})$. Звернемо увагу, що ініціалізація векторів W_i та w_i узгоджується з формулами (6) та (7).

1. *Пошук мета-нейрона-переможця.* Покладемо $k \leftarrow k + 1$. Знаходимо мета-нейрон-переможця i^* для поточного вхідного вектора $x_k \in X$, використовуючи формулу (1).

2. *Розширення мета-нейрона.* Розширюємо мета-нейрон i^* відповідно до формул (5), (6) та (7). У результаті отримуємо нейрон $i^{*ex}(w_{i^{*ex}}, W_{i^{*ex}}, S_{i^{*ex}})$.

3. *Перевірка інформаційної насиченості.* Обчислюємо інформаційну насиченість нейронів i^* та i^{*ex} . Якщо умова (8) виконується хоча б для одного виміру $l = \overline{1, n}$, то переходимо на наступний крок, інакше – крок 5.

4. *Розщеплення мета-нейрона.* Утворюємо новий мета-нейрон i^{new} , вектори якого ініціалізуються відповідно до процесу розщеплення: $w_{i^{new}} = (0, \dots, 0)$, $W_{i^{new}} = ([0, 0], \dots, [0, 0])$, $S_{i^{new}} = (\emptyset, \dots, \emptyset)$. Для кожного виміру $l = \overline{1, n}$, для якого умова (8) виконується та $x_k^l \in W_{i^*}^l$, розбиваємо множину $S_{i^*}^l$ і

утворюємо дві підмножини $S_{i,ex}^l$, $S_{i,new}^l$ відповідно до формул (9) та (10). Інакше застосовуємо формули (11) та (12). Мета-нейрон i^{new} додаємо до нейромережі.

5. *Перевірка завершення фази впорядкування.* Якщо $k < K$, то переходимо на крок 1.

6. *Ініціалізація етапу взаємодії.* Покладаємо $i \leftarrow 1$. Уведемо лічильник об'єднань мета-нейронів $m \leftarrow 0$.

7. *Пошук топологічного сусіда.* Використовуємо співвідношення (13) для пошуку найближчого мета-нейрона сусіда i^{bmi} для мета-нейрона i .

8. *Об'єднання.* Створюємо новий мета-нейрон \check{i} , об'єднуючи мета-нейрони i та i^{bmi} . Для цього застосовуємо формули (14), (6) та (7).

9. *Перевірка якості об'єднання.* Якщо умова (15) виконується при всіх $l = \overline{1, n}$, то об'єднання вважаємо вдалим: мета-нейрон \check{i} вносимо в нейромережу, а мета-нейрони i та i^{bmi} видаляємо й покладаємо $m \leftarrow m + 1$. Інакше нейромережа не змінюється, а мета-нейрон \check{i} відкидається як непридатний.

10. *Продовження.* Покладаємо $i \leftarrow i + 1$. Якщо $i < N$, то перехід на крок 7. Інакше – якщо $i \geq N$ і $m > 0$, то перехід на крок 6. Інакше – якщо $i \geq N$ і $m = 0$, то кінець.

У запропонованому алгоритмі для реалізації методу ADD кроки 1–5 відповідають фазі впорядкування, яка складається з етапів змагання та адаптації. Кроки 6–10 відповідають фазі консолідації, а відтак етапу взаємодії.

Приклади застосування нейромережі ADD. Порівняння з SOM

З метою перевірки дієвості запропонованої топології нейромережі та методу її навчання ADD було розроблено програмний нейросимулятор, із використанням якого було проведено низку комп'ютерних експериментів. У цьому розділі подано кінцеві результати навчання нейромережі ADD для двох тестових множин $X_q \subset R^2$, $q = \overline{1, 2}$. Зазначимо, що використання даних саме із двовимірного простору зумовлено легкістю візуального представлення результатів.

Отримані результати застосування навченої нейромережі ADD наводяться порівняно з результатами SOM, навченої методом Кохонена на тих самих множинах даних. Параметри для навчання SOM вибирались відповідно до рекомендацій із праці [5]. Для кожної із вхідних множин SOM складалась із двовимірної прямокутної ґратки. Для проведення найбільш адекватного порівняння кількість нейронів у кожному випадку вибиралась такою ж, як і кількість нейронів у вже навченій нейромережі ADD. Для обох множин навчання SOM методом Кохонена становило 10000 ітерацій.

З метою порівняння результатів застосування SOM та ADD використано два критерії: якість апроксимації вхідних даних та якість відображення структури вхідних даних (topology preservation). Якість апроксимації вимірюється як *середнє квадратичне відхилення (mean square error, MSE)* вхідних векторів та вагових векторів відповідних їм нейронів-переможців, яке обчислюється співвідношенням:

$$MSE = \frac{1}{|X|} \sum_{x \in X} \|x - w_c\|^2 \quad (16)$$

Оскільки моделі штучного нейрона в SOM та мета-нейрона в ADD володіють ваговим вектором, то співвідношення (16) може бути застосованим до обох нейромереж, що забезпечує уніфіковану характеристику для порівняння двох нейромереж.

Для виміру якості відображення структури вхідної множини у випадку SOM розроблено відповідні співвідношення [12]. Для ADD ці співвідношення неможливо застосувати, і на поточному етапі розробки нейромережі ADD аналогічних співвідношень ще не було розроблено. З огляду на це, якість відображення структури вхідної множини запропоновано перевіряти, візуально спостерігаючи за графічним зображенням вагових векторів нейронів у SOM та інтервальних векторів з ADD.

Тестова множина X_1 . Ця множина є простішою з обговорених і була використана з метою представлення базових можливостей адаптації нейромережі ADD. На рис. 5 зображено тестову

множину X_1 . Вона складається із двох підмножин квадратної форми, які в сукупності налічують 2000 рівномірно розподілених точок, тобто $|X_1| = 2000$.

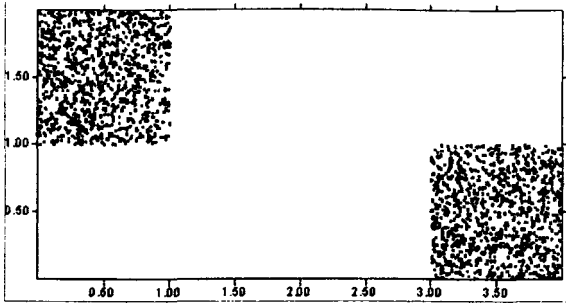


Рис. 5. Тестова множина X_1

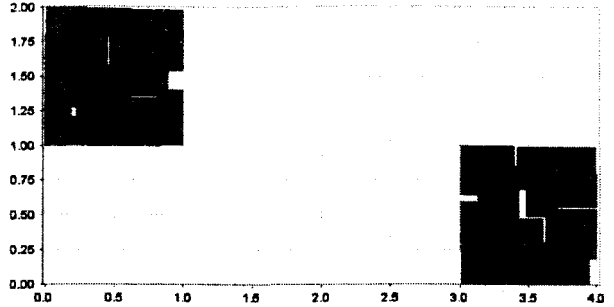


Рис. 6. Представлення множини X_1 нейромережею ADD

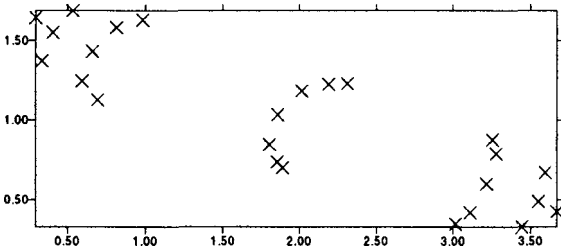


Рис. 7. Представлення множини X_1 нейромережею SOM

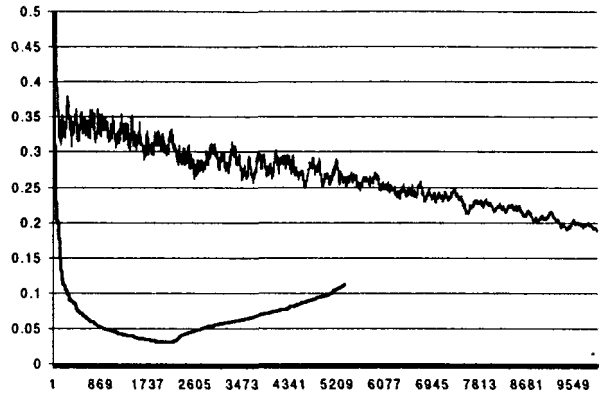


Рис. 8. Графіки зміни MSE в процесі навчання нейромереж SOM (осцилюючий) та ADD (плавний) на множині X_1

На рис. 6 зображено результат апроксимації множини X_1 навченою нейромережею ADD. Кожний сірий прямокутник відповідає інтервальному вектору W_i конкретного мета-нейрона $i = \overline{1, N}$, де N – кількість мета-нейронів у навченій нейромережі. Оскільки вхідні дані є двовимірними, то $W_i = (W_i^1, W_i^2)$, де $W_i^1 = [\underline{w}_i^1, \overline{w}_i^1]$ та $W_i^2 = [\underline{w}_i^2, \overline{w}_i^2]$. Використовуючи $|W_i^1|$ за ширину, а $|W_i^2|$ за висоту, можна побудувати прямокутник, верхня ліва вершина якого має координати $(\underline{w}_i^1, \overline{w}_i^2)$. Застосовуючи такий підхід до всіх мета-нейронів $i = \overline{1, N}$ отримаємо інтервальне покриття вхідної множини, зображене на рис. 6. Зазначимо, що темно-сірі області на рис. 6 відповідають зонам перекривання прямокутників. Після завершення навчання на множині X_1 нейромережа ADD містила 25 мета-нейронів.

З огляду на двовимірність вхідних даних вагові вектори нейронів у SOM є також двовимірними. Кількість нейронів у SOM становила 25 (ґратка 5×5), що відповідає кількості мета-нейронів у навченій ADD. На рис. 7 зображено вагові вектори нейронів у навченій на множині X_1 SOM. Аналізуючи рис. 5, 6 та 7, легко переконатись, що якість відображення структури вхідної множини нейромережею ADD є вищою, що автоматично забезпечує кращу кластеризацію. Зазначимо, що нейромережа SOM може забезпечити кращий результат, ніж представлено на рис. 7, але лише за більшої кількості нейронів.

На рис. 8 зображено графіки зміни MSE для процесів навчання нейромереж SOM та ADD. Вісь OX відповідає кількості ітерацій процесу навчання, а вісь OY відповідає значенням MSE. Осцилюючий графік, який розташований вище, відповідає процесу навчання нейромережі SOM. Плавний графік, розташований нижче, відповідає процесу навчання нейромережі ADD. Із представлених графіків легко переконатись, що навчання нейромережі ADD проходило рівномірніше, ніж нейромережі SOM, про що свідчить плавність графіку. Як уже обговорювалось раніше в цій статті, метод ADD складається із двох фаз: упорядкування та консолідації. Впродовж перших 2000 ітерацій графік, що відповідає процесу навчання методом ADD, спадає, а на подальших ітераціях – зростає. Частина графіка, яка характеризується спадом, відповідає фазі впорядкування, впродовж якої кількість мета-нейронів збільшується. Частина графіка, яка характеризується зростанням, відповідає фазі консолідації, впродовж якої кількість мета-нейронів зменшується, що призводить до деякого погіршення якості апроксимації вхідного простору. Остаточо, для SOM значення MSE становило 0.188619331, а для ADD – 0.11225949, забезпечуючи кращу апроксимацію вхідних даних.

Тестова множина X_2 . Множина X_2 має форму спіралі (див. рис. 9) і є структурно значно більш складною, ніж множина X_1 . Потужність множини є $|X_2| = 4963$ точок. Вектори множини X_2 є рівномірно розподілені.

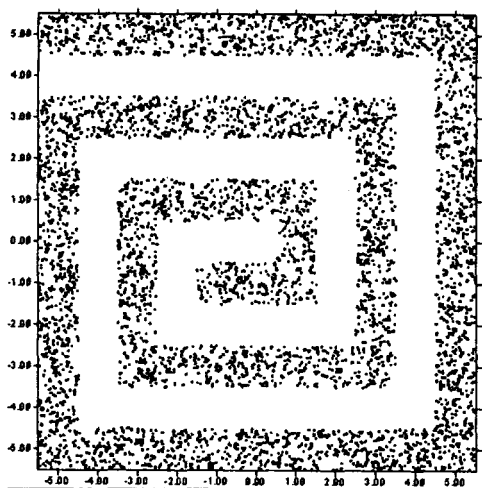


Рис. 9. Тестова множина X_2

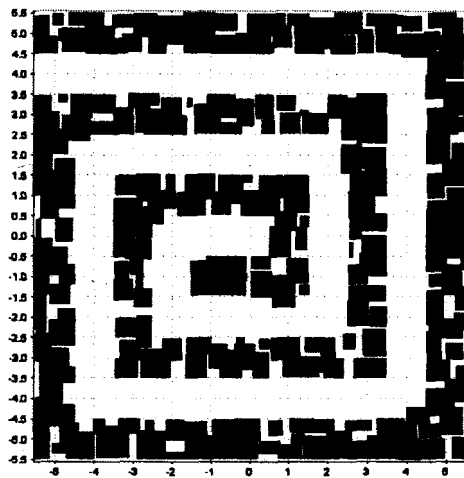


Рис. 10. Представлення множини X_2 нейромережею ADD

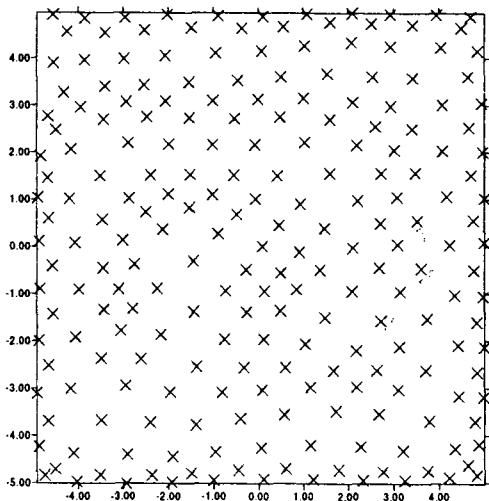


Рис. 11. Представлення множини X_2 нейромережею SOM

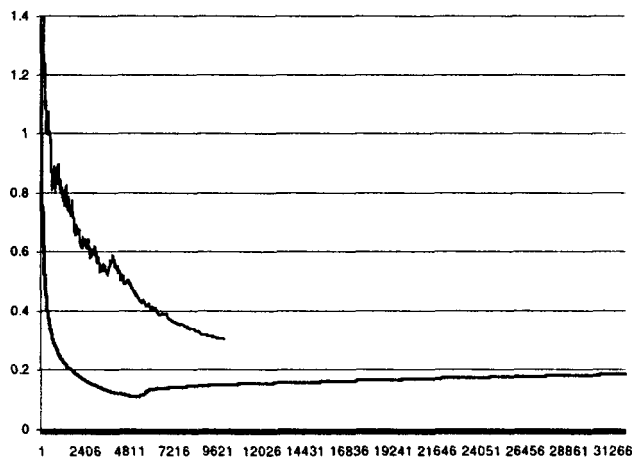


Рис. 12. Графіки зміни MSE в процесі навчання нейромереж SOM (осцилюючий) та ADD (плавний) на множині X_2

На рис. 10 зображено результат апроксимації множини X_1 навченою нейромережею ADD. Візуально легко перекопатись, що навчена ADD чітко відображає структуру спіралі, причому жоден із мета-нейронів не містить точок із різних смуг спіралі. Кількість мета-нейронів у навченій на множині X_2 нейромережі ADD становила 224.

На рис. 11 зображено вагові вектори нейронів у навченій на множині X_2 SOM. Аналізуючи рис. 11 легко перекопатись, що навчена SOM не забезпечує адекватного відображення структури спіралі. Використана SOM складалась з 225 нейронів, що відповідає ґратці 15×15 .

Графіки зміни *MSE* для процесів навчання нейромереж SOM та ADD на множині X_2 зображено на рис. 12. Як і у випадку із множиною X_1 , осцилюючий графік відповідає процесу навчання нейромережі SOM, а плавний графік – процесу навчання нейромережі ADD. Проходження процесу навчання для обох нейромереж значною мірою повторює проходження процесу навчання на множині X_1 за однієї відмінності – метод ADD характеризується в 5 разів більшою кількістю ітерацій (53000), ніж метод Кохонена. Звернемо увагу, що на рис. 12 відображено лише близько 32000 перших ітерацій, решта ітерацій незначно впливали на якість апроксимації й не були відображені для спрощення графіків. Із графіка *MSE* на рис. 12, який відповідає методу ADD, видно, що основна кількість ітерацій припадає на фазу консолідації. Зазначимо, що на даному етапі розробки нейромережі ADD неможливо порівнювати тривалість навчання методами ADD та Кохонена, оскільки тривалості однієї конкретної ітерації для кожного з методів відмінні. Ситуація додатково ускладнюється тим, що для самого методу ADD тривалість ітерації є різною для його різних фаз. Остаточо, для SOM значення *MSE* склало 0.306056497, а для ADD – 0.216015604 забезпечуючи кращу апроксимацію вхідних даних.

Динаміка розвитку нейромережі ADD. З огляду на те, що нейромережа ADD динамічно розвивається впродовж навчання, цікавим є зображення динаміки зміни нейромережі ADD впродовж навчання. Для цього можна представити графіки зміни кількості мета-нейронів на різних ітераціях методу ADD.

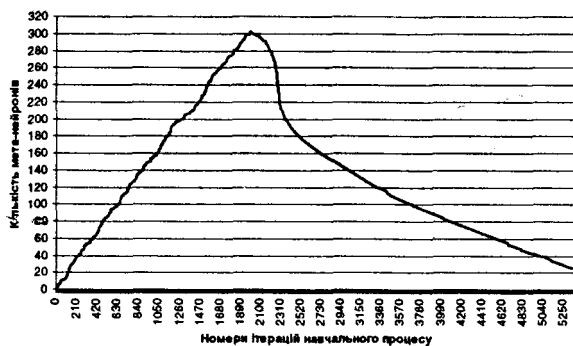


Рис. 13. Графік динаміки зміни кількості мета-нейронів у нейромережі ADD впродовж навчального процесу на множині X_1

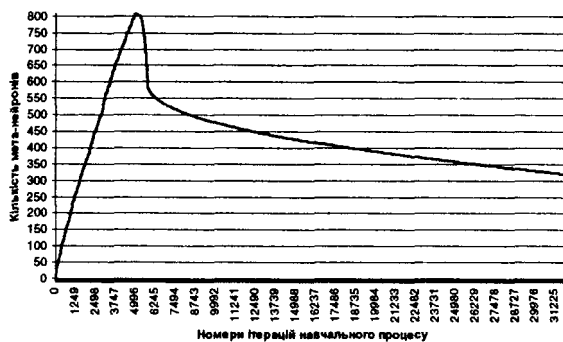


Рис. 14. Графік динаміки зміни кількості мета-нейронів у нейромережі ADD впродовж навчального процесу на множині X_2

Рис. 13 та 14 ілюструють динаміку зміни кількості мета-нейронів впродовж навчання для обговорених навчальних множин X_1 та X_2 . Кожен із представлених графіків можна умовно розділити на дві частини: зростання та спад. Зростання відповідає фазі впорядкування, а спад – фазі консолідації. Можна також спостерігати, що кількість ітерацій, впродовж яких відбувається зростання графіків, відповідає кількості елементів у відповідних навчальних множинах. Кількість ітерацій, які припадають на спад, безпосередньо не корелюється з потужністю навчальних множин, але залежить від кількості мета-нейронів, породжених на фазі впорядкування.

Зазначимо, що для фази впорядкування завжди достеменно відома кількість ітерацій, оскільки вона дорівнює кількості елементів у навчальній множині. Водночас, для етапу консолідації неможливо наперед визначити кількість ітерацій. Легко перекопатись із представлених графіків, що фаза консолідації є більш тривалою в усіх випадках.

Висновки

У цій статті обговорено перший етап дослідження з розробки нової топології нейромережі, яка покликана вдосконалити SOM для подолання недоліків, притаманних її топології. Вступна частина звертає увагу на основні з них, але більш детальне обговорення можна знайти в працях [6] та [7].

У проведеному дослідженні запропоновано нову модель штучного нейрона – мета-нейрон. Основною особливістю цієї моделі є здатність окреслити гіперкубічну область у вхідному просторі, за представлення даних з якої відповідає конкретний мета-нейрон. Нейромережа ADD, яка складається з мета-нейронів, реалізує розбиття вхідних даних на гіперкубічні області. Це забезпечує більш надійне представлення даних, ніж при “точковій” апроксимації, яка притаманна SOM та іншим її модифікаціям.

Для навчання нейромережі ADD було запропоновано однойменний метод ADD. За основу цього методу покладено концепцію інформаційної насиченості мета-нейронів та динамічної розбудови нейромережі, що повністю вирішує проблему в потребі “ручного” визначення структури нейромережі. Для визначення інформаційної насиченості було розроблено функцію ADD, яка задається співвідношеннями (3) та (4). Звернемо увагу, що запропонована функція ADD не вимагає задання жодних вхідних параметрів і ґрунтується виключно на множині даних, які представлені конкретним мета-нейроном. Проведені експерименти не виявили вагомих недоліків такого підходу. Хоча зауважимо, що вимагається проведення додаткових комп’ютерних симуляцій для остаточного ствердження, що запропонована функція ADD є ефективною в усіх випадках. Зокрема, такі симуляції передбачають використання навчальних множин, які містять дані з різними розподілами.

Важливо зазначити, що додатковою перевагою (яка не обговорюється в цій статті) моделі мета-нейрона в сукупності з методом ADD є легкість донавчання нейромережі, що є надзвичайно важливою властивістю на практиці. Процес навчання будь-яким методом може бути ресурсомістким, що зумовлено низкою чинників – від складності вхідних даних до особливостей обчислення конкретного навчального методу. Тому під час зміни вхідних даних, а це в першу чергу стосується внесення нових даних, які потенційно можуть формувати нові кластери, важливо мати можливість донавчання існуючої нейромережі. Тобто, продовжити навчання з використанням уже навченої нейромережі, а не починати процес із самого початку, що за значних кількісних змін вхідних даних може призвести до критичного зростання часу, який вимагається для навчання. Яскравим прикладом такої проблеми є метод Кохонена навчання SOM, де процес навчання базується на концепції топологічного сусідства нейронів [5], яке модифікується впродовж навчання. Доновчання в цьому випадку означатиме перенавчання нейромережі з фактичною втратою попередньо набутого досвіду.

З огляду на модель мета-нейрона, яка дає змогу встановити приналежність вхідних даних до конкретного мета-нейрона, використовуючи його інтервальний вектор W , та здатність методу ADD до динамічної розбудови нейромережі, процес донавчання може бути ефективно реалізовано. Одним із завдань наступного етапу дослідження нейромережі ADD та методу навчання ADD є реалізація та оцінка ефективності донавчання.

Залишається відкритим питання виявлення та представлення мета-нейронів у навченій нейромережі ADD, які відповідають за представлення вхідних даних із різних кластерів. Одним із важливих напрямків подальшого розвитку представленого дослідження є розробка підходу до маркування мета-нейронів саме з цією ціллю. Одним із можливих підходів є групування мета-нейронів у навченій нейромережі за умови перетину їхніх гіперкубів, які представлені інтервальними ваговими векторами. Мета-нейрони, які не мають перетину, вважатимуться такими, що відповідають за представлення різних кластерів.

Подальше дослідження нейромережі та методу ADD передбачає розробку уніфікованих мір якостей навчання та застосування нейромережі для ефективного порівняння запропонованої технології із існуючими нейромережевими технологіями, які спрямовані на розв’язування задач кластеризації.

1. Kriventseva E., Biswas M., Apweiler R. *Clustering and analysis of protein families. Current Opinion in Structural Biology*, 11, pp. 334 – 339, 2001. 2. Linial M., Linial N., Tishby J., Golan Y. *Global self-organization of all known protein sequences reveals inherent biological structures. Journal of Molecular Biology*, 268, pp. 539556, 1997. 3. Miam I., Dubchak I. *Representing and reasoning about protein families using generative and discriminative methods. Journal of Computational Biology*, 7(6), pp. 849 – 862, 2000 4. Kohonen T. *Self-organized Formation of Topologically Correct Feature Maps.*

Biological Cybernetics, 43, pp. 59-69, 1982. 5. Haykin S. *Neural Networks: A comprehensive foundation*, Prentice-Hall, New Jersey, 1999. 6. Si J., Lin S., Vuong M.-A. *Dynamic topology representing network*, *Neural Networks Vol 13*, pp. 617-627, 2000. 7. Fritzke B. *Growing cell structure – a self-organising network for unsupervised and supervised learning*, *Neural Networks, Vol. 7 (9)*, pp. 1441-1460, 1995. 8. Blackmore, J., *Visualizing high-dimensional structure with the incremental grid growing neural network. Thesis*, 1995, www.cs.utexas.edu/users/nn/downloads/papers/blackmore.thesis.pdf 9. Bauer H., Villman T. *Growing a hypercubical output space in a self-organizing feature map*. *IEEE Transactions on Neural Networks*, 8, pp. 218-226, 1997. 10. Годич О. Навчання SOM методом нейронної міграції // Вісник НУ “Львівська політехніка”. – 2004. – № 519. – С. 55-72. 11. Hansen E., Walster G. *Global optimization using interval analysis*, Second Edition, Martkel Dekkel, Inc. New York, 2004. 12. Osowski S. *Sieci Neuronowe w ujeciu algorytmicznym*. Wydawnictwo Naukowo-Techniczne, Warszawa, 1996.

УДК 683.1

А.Г. Григорович*, В.Г. Григорович

*Дрогобицький педагогічний ліцей
Дрогобицький державний педагогічний університет імені Івана Франка,
кафедра інформаційних систем і технологій

НЕНОРМАЛІЗОВАНІ ВІДНОШЕННЯ: ІСТОРІЯ, КОНЦЕПЦІЇ ТА ТЕНДЕНЦІЇ РОЗВИТКУ. ЧАСТИНА 1. БАЗОВІ ПОНЯТТЯ, КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ, МОДЕЛЬ ВЕРСО

© Григорович А.Г., Григорович В.Г., 2005

Наведено першу частину дослідження, в якому проаналізовано історію, основні концепції та тенденції розвитку ненормалізованих відношень. Розглянуто модель Версо та реляційні операції над ненормалізованими відношеннями. Запропоновано розширення моделі “Сутність–Зв’язок” для випадку складних та вкладених сутностей.

The article is the first part of the research in which the authors analyze history, main conceptions and trends of development of nested relations. There are examined Verso model and relational operations upon nested relations. The authors also propose the extension of Entity-Relationship model for cases of nested entities.

Вступ. Загальна постановка проблеми

Проблема адекватного подання даних в інформаційних системах не перестає бути актуальною: з одного боку – для існуючих задач будують нові моделі даних, які адекватніше подають предметну галузь; з іншого боку – виникають нові задачі, які потребують опрацювання даних нової природи. Композитні класи сутностей – ті, які містять сутності інших класів, – зустрічаються надзвичайно часто. Зокрема, в класичній задачі складського обліку сутності *Накладна* та *Замовлення* містять сутності *Відпущений_Товар* та *Замовлений_Товар* відповідно. У межах класичної моделі “Сутність–Зв’язок” вкладені сутності подають за допомогою слабких класів, відношення яких містять зовнішні ключі для зв’язку із відношеннями відповідних сильних класів. Під час переходу від ER-діаграми до реляційної схеми в класичній реляційній моделі втрачається відмінність між відношенням, побудованим на основі сильного класу, та відношенням, побудованим на базі слабого класу сутностей. Отже, класична реляційна модель даних не є адекватною в надзвичайно широкому спектрі застосувань. Ненормалізовані відношення, ідея яких виникла в 1977 році [1], успішно вирішують проблему подання вкладених сутностей.