ЗАСТОСУВАННЯ АНАЛІТИЧНОЇ ПЛАТФОРМИ DEDUCTOR ПРИ ВИВЧЕННІ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА ЕКСПЕРТНІ СИСТЕМИ В МІЖНАРОДНИХ ВІДНОСИНАХ"

Заверач М.М.

доцент кафедри міжнародної інформації
Хмельницького національного університету

При вивченні навчальної дисципліни "Системи штучного інтелекту та експертні системи в міжнародних відносинах" на кафедрі міжнародної інформації Хмельницького національного університету на протязі багатьох років широко застосовується аналітична платформа Deductor починаючи із 3 версії. З 2010 року розпочато освоєння Deductor Academic версії 5.2. При викладанні матеріалу передбачені лекційні та лабораторні заняття а також самостійна робота. На лабораторних роботах студенти вивчають нейроні мережі, карти Кохонена та часові ряди, і здійснюють прогнозування на основі даних, які приведені в джерелах інформації [1-3]. Самостійна робота передбачає застосування отриманих знань та навиків для отримання прогнозів у міжнародних відносинах.

Використання нейронних мереж є одним із перспективних кількісних методів прогнозування. Важливою перевагою нейронних мереж є їх гнучка структура. Для зміни структури у рамках визначеної архітектури нейронної мережі достатньо регулювати кількість шарів та нейронів, додаткові переваги надає також можливість зміни активаційної функції. Такі перетворення надають можливість повністю змінити структуру мережі, що дозволяє максимально пристосувати обрану архітектуру і мінімізувати похибку навчання мережі та підвищити точність прогнозування. Використовуючи ж навіть найпростішу нейромережеву архітектуру (персептрон з одним прихованим шаром) і базу даних (із інформацією про минулі події) легко одержати працюючу систему прогнозування. Ще одна серйозна перевага нейронних мереж полягає в тому, що експерт не залежить від вибору математичної моделі поведінки часового ряду. Побудова нейромережевої моделі відбувається адаптивно під час навчання, без участі експерта. При цьому нейронній мережі надаються приклади з бази даних і вона сама налагоджується під ці дані.

Недоліком нейронних мереж є їхня недетермінованість. Мається

на увазі те, що після навчання мережа є "чорним ящиком", який якимсь чином працює, але логіка прийняття рішень нейронною мережею прихована від експерта.

Аналітична платформа Deductor містить повний набір механізмів імпорту, обробки, візуалізації й експорту даних для швидкого й ефективного аналізу інформації. Робота з аналізу даних у Deductor базується на виконанні наступних дій: імпорт даних; обробка даних; візуалізація та експорт даних. Усі механізми уніфіковані і виконуються за допомогою майстрів.

Відправною точкою для аналізу завжди є процедура імпорту даних. Отриманий набір даних може бути оброблений різними алгоритмами. Результати обробки можна переглянути різними способами і експортувати. Послідовність дій, які необхідно провести для аналізу даних називається сценарієм, який можна автоматично виконувати на будь-яких даних.

У Deductor використовуються такі технології як багатомірний аналіз, нейронні мережі, дерева рішень, самонавчальні карти та інші. При цьому акцент зроблений на самонавчальні методи і машинне навчання. Використання методів, що самонавчаються, і майстрів для настроювання, дозволяє знизити вимоги до підготовки персоналу, роблячи сучасні технології доступними широкому колу користувачів.

Для того, щоб почати аналіз, необхідно одержати табличні дані зі стороннього джерела. Deductor підтримує багато джерел даних: сховище даних Deductor Warehouse, промислові СУБД (Oracle, MS SQL), текстові файли, офісні програми (Excel, Access), ADO і ODBC джерела.

Наступним кроком є обробка даних. Під обробкою розуміють будь-яку дію, що зв'язана з перетворенням даних. Механізми обробки можна комбінувати довільним чином. Доступні наступні: нейронні мережі, дерева рішень, самонавчальні карти, асоціативні правила, лінійна регресія та інші.

Переглянути результати можна за допомогою механізмів візуалізації. Візуалізувати можна будь-який об'єкт у сценарії обробки. Програма самостійно аналізує, яким чином можна відобразити інформацію, користувач повинен тільки вибрати потрібний варіант – статистика, граф нейронної мережі, ієрархічна система правил, карти та інше.

Завершальним кроком у сценарії обробки є експорт даних. Підтримуються наступні формати: сховище даних Deductor Warehouse, Microsoft Access, Microsoft Excel, Microsoft Word, HTML, XML, Dbase,

текстовий файл із роздільниками.

Побудова нейронної мережі в платформі Deductor включає в себе декілька кроків. На першому кроці проводиться імпорт даних в Deductor. При потребі над даними можна виконати ряд операцій: очистку та трансформацію даних, парціальну та спектральну обробку, відновлення пропущених даних. Для побудови нейронної мережі слід запустити обробник "Нейронна мережа". В подальшому потрібно задати налаштування нейронної мережі. Вказати, які поля є вхідними, вихідними та інформаційними. Далі здійснюється розподіл всієї множини на навчальну та тестову. Значення, вказані по замовчуванню, можна залишити без змін. Потім задаються параметри, що визначають структуру нейронної мережі, – кількість прихованих шарів і нейронів в них, а також активаційна функція нейронів.

До вибору кількості прихованих шарів і кількості нейронів для кожного прихованого шару потрібно підходити обережно. Вважається, що задачу будь-якої складності можна вирішити за допомогою двохшарової нейронної мережі, тому конфігурація з кількістю прихованих шарів, що перевищують 2, навряд чи виправдана. Для вирішення багатьох завдань цілком підійде й одношарова нейронна мережа. При виборі кількості нейронів слід керуватися наступним правилом: "кількість зв'язків між нейронами повинна бути приблизно на порядок менше кількості прикладів в навчальній множині". Кількість зв'язків розраховується як сума зв'язків кожного нейрона зі всіма нейронами сусідніх шарів, включаючи зв'язки на вхідному і вихідному шарах. Дуже велика кількість нейронів може призвести до так званого "перенавчання" мережі, коли вона видає добрі результати на прикладах, що входять в навчальну вибірку, але практично не працює на інших прикладах.

Число нейронів у вхідному і вихідному шарах автоматично встановлюється відповідно до числа вхідних і вихідних полів навчальної вибірки і змінити його не можна.

Наступний крок – налаштування процесу навчання нейронної мережі. Можна вибрати два способи: Back-Propagation (зворотнє розповсюдження помилки) та Resilient Propagation (еластичне розповсюдження помилки). Кожен метод має по два параметри. Back-Propagation: 1) "Швидкість навчання" – визначає величину кроку при ітераційній корекції ваг в нейронній мережі (рекомендується задавати в інтервалі (0,1) та 2) "Момент" – враховує величину останньої зміни ваги при корекції ваг (задається в інтервалі (0,1). Resilient Propagation: 1)

"Крок спуску" – коефіцієнт збільшення швидкості навчання, який визначає крок збільшення швидкості навчання при недосягненні алгоритмом оптимального результату та 2) "Крок підйому" – коефіцієнт зменшення швидкості навчання, задається крок зменшення швидкості навчання в разі пропуску алгоритмом оптимального результату.

Далі вказується параметри, при виконанні яких навчання мережі буде зупинено. Після цього проводиться навчання нейронної мережі. Навчання вважатися успішним, якщо відсоток розпізнаних прикладів в навчальній та тестовій множинах достатньо великий (близький до 100%).

Приклад оцінки кредитоспроможності фізичної особи за допомогою нейронних мереж. Основні фактори, що обумовлюють кредитоспроможність: вік, освіта, площа квартири, наявність автомобіля, тривалість проживання в даному регіоні. Необхідна також статистика повернень або не повернень взятих кредитів. Приведені дані використаємо для прогнозування оцінки кредитоспроможності фізичної особи за допомогою нейронних мереж.

Для побудови нейронної мережі необхідно нормалізувати поля. Поля "Сума кредиту", "Вік", "Площа квартири" і "Тривалість проживання" – безперервні значення, які перетворимо до інтервалу [–1..1]. Освіту представлено трьома унікальними значеннями, які можна порівнювати на більше або менше, а точніше краще або гірше. Тобто освіту можна впорядкувати так: середня, спеціальна, вища. Значення поля з наявністю автомобіля впорядкувати не можна. Його потрібно перетворити до бітової маски. Для кодування трьох значень потрібно два біти. Отже, це поле буде розбито на два.

Навчальну вибірку розіб'ємо на навчальну й тестову множини так, як програма пропонує це зробити за замовчуванням. Тобто у навчальну множину потраплять випадкові 95 відсотків значень, а інші 5 відсотків – у тестову.

Конфігурація мережі буде такою. У вхідному шарі – 7 нейронів, тобто по одному нейрону на один вхід (у навчальній вибірці 6 стовпців, але стовпець "Автомобіль" представлений бітовою маскою із двох біт, для кожного з яких створений новий вхід). Зробимо один прихований шар із двома нейронами. У вихідному шарі буде один нейрон, на виході якого буде рішення про видачу кредиту. Виберемо алгоритм навчання мережі – Resilient Propagation з настроюваннями за замовчуванням. Умову закінчення навчання залишимо без зміни.

Навчену в такий спосіб нейронну мережу можна використовувати

для прийняття рішення про видачу кредиту фізичній особі. Це можна зробити за допомогою аналізу "що-якщо". Для його включення потрібно вибрати візуалізатор "Що-якщо". Після зміни в цій таблиці значень вхідних полів система розраховує умови видачі кредиту й у полі "Давати кредит" проставляє "Так", або "Ні".

Самонавчальні карти Кохонена, – це один із різновидів нейронних мереж, які використовують неконтрольоване навчання. При такому навчанні навчальна множина складається лише із значень вхідних змінних, у процесі навчання немає порівняння виходів нейронів з еталонними значеннями. Можна сказати, що така мережа вчиться розуміти структуру даних.

Карти, що самі організуються (Self Organizing Maps) можуть використовуватися для вирішення таких задач як моделювання, прогнозування, пошук закономірностей у великих масивах даних, виявлення наборів незалежних ознак і стиснення інформації. Найпоширеніше застосування мереж Кохонена – рішення задачі класифікації без учителя, тобто кластеризації.

Перед початком навчання карти необхідно провести ініціалізацію вагових коефіцієнтів нейронів. Вдало обраний спосіб ініціалізації може суттєво прискорити навчання й привести до одержання більш якісних результатів. Існують такі способи ініціювання початкових ваг: ініціалізація випадковими значеннями, коли всім вагам присвоюються малі випадкові величини; ініціалізація прикладами, коли в якості початкових значень задаються значення випадково обраних прикладів з навчальної вибірки; лінійна ініціалізація. У цьому випадку ваги ініціюються значеннями векторів, лінійно впорядкованих уздовж лінійного підпростору, що проходить між двома головними власними векторами вихідного набору даних.

Отриману в результаті навчання карту можна представити у вигляді прошаркового пирога, кожний шар якого являє собою розфарбування, породжене однією із компонентів вихідних даних. Отриманий набір розфарбувань може використовуватися для аналізу закономірностей, наявних між компонентами набору даних.

Приклад оцінки кредитоспроможності фізичної особи за допомогою карт Кохонена. Для аналізу ринку кредитування необхідно в першу чергу зрозуміти загальну картину. Хто бере кредити, навіщо, які існують причини відмовлень у видачі кредитів або причини неспроможності. Для цього необхідне наочне представлення всіх наявних даних. Таку задачу можна вирішити за допомогою побудови

самонавчальних карт Кохонена.

Навчальну вибірку будемо застосовувати ту ж, що й для нейронної мережі. Розбивку навчальної вибірки на дві множини залишимо за замовчуванням, так само, як і інші настроювання. Наступні кроки пропонують настроїти параметри карти (кількість комірок по X і по Y, їхню форму) і параметри навчання (спосіб початкової ініціалізації, тип функції сусідства, чи перемішувати рядки навчальної множини й кількість епох, через які необхідне перемішування) та параметри кластеризації – автоматичне визначення числа кластерів з відповідним рівнем значимості або фіксована кількість кластерів. Надається також можливість настроїти інтервали навчання. Кожний інтервал задається кількістю епох, радіусом навчання й швидкістю навчання.

Після завершення процесу обробки необхідно в списку візуалізаторів вибрати "Карту Кохонена" для перегляду результатів кластеризації, а також візуалізатор "Що-якщо". Далі, у майстрові настроювання відображення карти Кохонена необхідно вказати, щоб відображалися всі поля і поставити прапорець "Границі кластерів". Після цього результати роботи алгоритму відображаються на картах, що показують розподіл позичальників по характеристиках "Сума кредиту", "Термін кредиту", "Ціль кредитування" (турпоїздки, покупка товарів, покупка та ремонт нерухомості, оплата навчання, оплата послуг, та інше), "Середньомісячний прибуток", "Кількість утриманців" і "Вік". У результаті кластеризації позичальники із схожими характеристиками потраплять в один кластер, і тому для них можна застосовувати однакові правила видачі кредиту, тобто для кожного кластера визначити, чи доцільно видавати кредит його представникам.

Прогнозування результату на певний час уперед, ґрунтуючись на даних за минулий час (часовий ряд), – завдання, що зустрічається досить часто. Наприклад, перед більшістю торговельних фірм стоїть завдання оптимізації складських запасів, для вирішення якої потрібно знати, що й скільки повинно бути продано через тиждень і т.п., завдання прогнозування вартості акцій якого-небудь підприємства через день і т.д. і інші подібні питання. Deductor пропонує для цього інструмент "Прогнозування", який з'являється в списку Майстра обробки тільки після побудови моделі прогнозу. Прогнозувати на кілька кроків уперед має сенс тільки часовий ряд (наприклад, якщо є дані по тижневих сумах продажів за певний період, можна спрогнозувати суму продажів на два тижні вперед).

В аналітика є дані про помісячну кількість проданого товару за

декілька років. Йому необхідно, ґрунтуючись на цих даних, визначити, яка кількість товару буде продана через місяць чи через два. Після імпорту даних необхідно скористатися діаграмою для їхнього перегляду. Якщо дані містять аномалії (викиди) і шуми, за якими важко розглянути тенденцію, то перед прогнозуванням необхідно видалити аномалії й згладити дані. Зробити це можна за допомогою парціальної обробки. Запустимо Майстер обробки, виберемо в якості обробки даних парціальну обробку. Наступний крок відповідає за видалення аномалій з вихідного набору. Виберемо поле для обробки "Кількість" і вкажемо для нього обробку аномальних явищ (ступінь придушення – мала). Четвертий крок Майстра дозволяє провести спектральну обробку. З вихідних даних необхідно виключити шуми, тому вибираємо стовпець "Кількість" і вказуємо спосіб обробки "Вирахування шуму" (ступінь вирахування – мала). Після обробки необхідно перевірити отриманий результат за допомогою діаграми.

Тепер перед аналітиком встає питання, а як, властиво, прогнозувати часовий ряд. Будувати прогноз на майбутнє необхідно, ґрунтуючись на даних минулих періодів, тобто припускаючи, що кількість продажів на наступний місяць залежить від кількості продажів за попередні місяці. Це значить, що вхідними факторами для моделі можуть бути продажі за поточний місяць, продажі за місяць раніше і т.д., а результатом повинні бути продажі за наступний місяць, тобто тут явно необхідно трансформувати дані до вікна що ковзає.

Запустимо Майстер обробки, виберемо в якості оброблювача вікно що ковзає й перейдемо на наступний крок. Необхідно виявити наявність річної сезонності за допомогою авторегресійного аналізу. Будуємо прогноз на місяць уперед, ґрунтуючись на даних за 1, 2, 11 і 12 місяці тому. Вибираємо глибину занурення 12, призначивши поле "Кількість" використовуваним. Тоді дані трансформуються до вікна що ковзає так, що аналітикові будуть доступні всі потрібні фактори для побудови прогнозу. Тепер у якості вхідних факторів можна використовувати "Кількість – 12", "Кількість – 11" – дані по кількості 12 і 11 місяців тому (щодо прогнозованого місяця), а також "Кількість – 2" і "Кількість – 1" – дані за 2 попередніх місяці. У якості вихідного поля вкажемо стовпець "Кількість".

Далі необхідно приступити до побудови моделі прогнозу. В Майстрові обробки виберемо нейронну мережу. На другому кроці Майстра встановимо в якості вхідних поля "Кількість – 12", "Кількість – 11", "Кількість – 2" і "Кількість – 1", а в якості вихідного – "Кількість".

На наступному кроці вкажемо розбивку тестової й навчальної множин. Перейдемо до наступного кроку, на якому відзначимо необхідну кількість шарів і нейронів у нейронній мережі. Далі, необхідно вибрати алгоритм навчання нейронної мережі.

Після побудови моделі для перегляду якості навчання представимо отримані дані у вигляді діаграми й діаграми розсіювання. Діаграма розсіювання більш наочно показує якість навчання. Нейронна мережа навчена, залишилося одержати необхідний прогноз. Для цього відкриваємо Майстер обробки й вибираємо оброблювач "Прогнозування". Майстер сам вірно настроїть усі переходи, тому залишається тільки вказати горизонт прогнозу рівний трьом. Після цього необхідно в якості візуалізатора вибрати "Діаграму прогнозу", яка з'являється тільки після прогнозування часового ряду. Тепер аналітик може дати відповідь на запитання, яка кількість товарів буде продана в наступному місяці й навіть через два місяці.

Даний приклад показав, як за допомогою Deductor Studio прогнозувати часовий ряд. При рішенні завдання були застосовані механізми очищення даних від шумів, аномалій, які забезпечили якість побудови моделі прогнозу й відповідно достовірний результат самого прогнозування кількості продажів на три місяці вперед. Також був продемонстрований принцип прогнозування часового ряду – імпорт, виявлення сезонності, очищення, згладжування, побудова моделі прогнозу й властиво побудова прогнозу часового ряду. Подібний сценарій – основа будь-якого прогнозування часового ряду з тою різницею, що для кожного випадку необхідно, як одержувати необхідний часовий ряд за допомогою інструментів Deductor, так і підбирати параметри очищення даних і параметри моделі прогнозу (наприклад, структуру мережі, якщо використовується навчання нейронної мережі, визначення значущих вхідних факторів).

1. *Документация по Deductor. [Електронний ресурс]. – Режим доступу: http://www.basegroup.ru/download/guides/ (дата звернення 05.05.2011).*
2. *Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям: Учеб. пособие – 2-е изд., доп. и перераб. – СПб.: Питер, 2010. – 701 с.: ил.*
3. *Кацко И. А., Паклин Н. Б. Практикум по анализу данных на компьютере: Учеб. пособие для вузов / Под ред. Гореловой Г.В. – М.: "КолосС", 2009. – 278 с.: ил.*