

Практична робота №7 «Аналітичні рішення за допомогою нейронних мереж».

Завдання до практичної роботи.

1. Відповідно до заданого варіанту підготувати навчальну вибірку у вигляді таблиць MS Excel і зберегти їх як персональні файли. Для підготовки даних використовувати тематичні сайти Інтернет, результати проходження практик, довідники і каталоги.

2. Провести нормалізацію полів, створення структури і навчання нейромережі в пакеті Deductor відповідно до методики, представленої в розділі «Порядок виконання роботи».

3. Отримати аналітичне рішення і провести декілька експериментів, використовуючи візуалізацію «Що - якщо» відповідно до методики, представленої в розділі «Порядок виконання роботи».

4. Виконати моделювання і економічну оцінку результатів шляхом використання діаграм відповідно до методики, представленої в розділі «Порядок виконання роботи».

6. Здійснити наступний імпорт даних з сховища: кількість відвантаженого товару в розрізі дат та товарів по вибраному Вами клієнтові, залишивши одну властивість товару (вибір властивості довільний). Виконання завдання здійснюється відповідно до методики, представленої в розділі «Порядок виконання роботи. Частина 4».

Варіанти завдань:

N варіанту	Процес
1	Діагностика захворювань
2	Діагностика комп'ютерів
3	Діагностика технічних систем
4	Діагностика комп'ютерних мереж
5	Прогнозування курсу валют
6	Прогнозування вартості нерухомості
7	Прогнозування курсу акцій
8	Оцінка реалізація медичних препаратів
9	Прогнозування рівня інфляції
10	Оцінка фінансового стану фірми
11	Оцінка кредитоспроможності фірми
12	Прогнозування рівня цін на продовольчі товари
13	Прогнозування рівня цін на промислові товари

Теоретична частина.

Нейронні мережі (НМ) є обчислювальними структурами, що моделюють прості біологічні процеси, аналогічні процесам, які відбуваються в

людському мозку. НМ – це розподілені і паралельні системи, здібні до адаптивного навчання шляхом реакції на позитивні і негативні дії. У основі побудови НМ лежить елементарний перетворювач, названий *штучним нейроном* або просто *нейроном* по аналогії з його біологічним прототипом.

Структуру нейромережі можна описати таким чином. Нейромережа складається з декількох шарів: вхідного, внутрішнього (прихованого) і вихідного шарів. Вхідний шар реалізує зв'язок з вхідними даними, вихідний – з вихідними. Внутрішніх шарів може бути від одного і більше. У кожному шарі міститься декілька нейронів. Між нейронами є зв'язки, названі *вагами* (рис. 7.1).

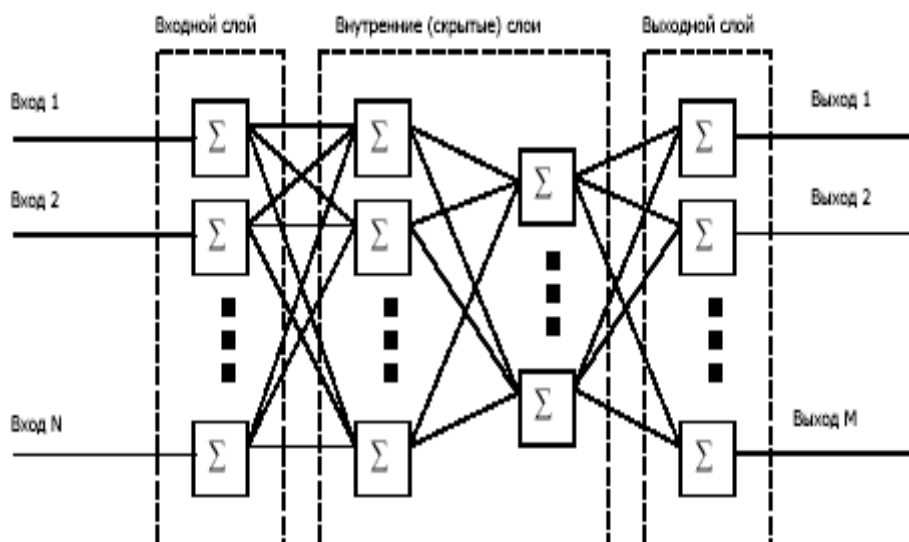


Рис. 7.1. Структура нейромережі.

Призначення і підготовка навчальної вибірки. Нейромережа здатна імітувати який-небудь процес. Будь-яка зміна входів нейромережі веде до зміни її виходів. Причому виходи нейромережі однозначно залежать від її входів. Але перш ніж використовувати нейромережу її необхідно навчити. Задача навчання тут рівносильна задачі апроксимації функції, тобто відновлення функції по окремо взятих її точках – таблично заданій функції. Таким чином, для навчання *потрібно підготувати таблицю* з вхідними значеннями і відповідними їм вихідними значеннями, тобто підготувати *навчальну вибірку*. По такій таблиці нейромережа сама знаходить залежності вихідних полів від вхідних. Далі ці

залежності можна використовувати, подаючи на вхід нейромережі деякі значення. На виході будуть відновлені залежні від них значення. Причому на вхід можна подавати значення, на яких нейромережа не навчалася.

Після навчання на вхід нейромережі необхідно подавати значення з діапазону, на якому вона навчалася. Наприклад, якщо при навчанні нейромережі на один з її входів подавалися значення від 0 до 100, то надалі слід на цей вхід подавати значення з діапазону від 0 до 100. Допускається подавати значення, які лежать поряд з діапазоном. На самому початку роботи з нейромережею потрібно визначитися, що є її входами, а що – виходами. Передбачається, що у нас вже є таблиця з навчальною вибіркою.

Нормалізація значень полів. Після того, як вказані вхідні і вихідні поля, слід нормалізувати дані в навчальній вибірці. Метою нормалізації значень полів є перетворення даних до вигляду, найбільш відповідного для обробки засобами Deductor Studio. Для таких обробників як нейронна мережа, дерево рішень, лінійна модель прогнозування дані, що поступають на вхід, повинні мати *числовий тип*, а їх значення мають бути розподілені в певному діапазоні.

Нормалізатор може перетворити дискретні дані до набору унікальних індексів або значень в діапазоні $[0...1]$. Для нейромережі доступні наступні види нормалізації полів.

1. **Лінійна нормалізація.** Використовується лише для безперервних числових полів. Дозволяє привести числа до діапазону $[\min...max]$, тобто мінімальному числу з вихідного діапазону відповідатиме $\min$, а максимальному – $\max$. Останні значення розподіляться між $\min$ та $\max$.

Унікальні значення. Використовується для дискретних значень. Такими є рядки, числа або дати, задані дискретно. Аби привести безперервні числа в дискретні можна, наприклад, скористатися обробкою «квантування». Так слід поступати з величинами, для яких можна задати відношення порядку, тобто, якщо для двох будь-яких дискретних значень можна вказати, яке більше, а яке менше. Тоді всі значення необхідно розташувати в порядку зростання. Далі вони нумеруються по порядку і значення замінюються їх порядковим номером.

3. Бітова маска. Використовується для дискретних значень. Цей вигляд нормалізації слід використовувати для величин, які можна лише порівнювати на рівність або нерівність, але не можна сказати, яке більше, а яке менше. Всі значення замінюються порядковими номерами, а номер розглядається в двійковому вигляді або у вигляді маски з нулів і одиниць. Тоді кожна позиція маски розглядається як окреме поле, що містить нуль або одиницю. До такого поля можна застосувати лінійну нормалізацію, тобто замінити нуль на деяке мінімальне значення, а одиницю – на деяке максимальне значення. Після такої нормалізації на вхід нейромережі подаватиметься не одне це поле, а стільки полів, скільки розрядів в масці.

Існує налаштування нормалізації полів за умовчанням. Тобто, запустивши обробник, в даному випадку нейронну мережу, цей етап можна пропустити. В цьому випадку нормалізація буде виконана автоматично залежно від типів і видів полів:

1. вигляд дискретний – нормалізація списком унікальних значень;
2. вигляд безперервний
 - тип цілий або дата – лінійна нормалізація в діапазоні $[-1..1]$;
 - тип вещественный - лінійна нормалізація в діапазоні $[0..1]$.

Налаштування навчальної вибірки. Після нормалізації полів слід налаштувати навчальну вибірку. Навчальну вибірку розбивають на дві множини – *навчальну* і *тестову*.

Навчальна множина - включає записи (приклади), які використовуватимуться безпосередньо для навчання мережі, тобто міститимуть вхідні і бажані вихідні (цільові) значення.

Тестова множина - також включає записи (приклади), що містять вхідні і бажані вихідні (цільові) значення, але використовується не для навчання мережі, а для перевірки результатів навчання.

Розбивати навчальну вибірку на ці множини можна двома способами: або по порядку, або випадково. Якщо розбиття відбувається по порядку, то тестова множина вибирається або з початку, або з кінця навчальної вибірки.

Далі задаються параметри, що визначають структуру нейронної мережі, - кількість прихованих шарів і нейронів в них, а також активаційна функція нейронів.

Зауваження. До вибору кількості прихованих шарів і кількості нейронів для кожного прихованого шару потрібно підходити обережно. Хоча до цих пір не вироблені чіткі критерії вибору, дати деякі загальні рекомендації все ж можливо. Вважається, що задачу будь-якої складності можна вирішити за допомогою двошарової нейромережі, тому конфігурація з кількістю прихованих шарів, що перевищують 2, навряд чи виправдана. Для вирішення багатьох задач сповна підійде одношарова нейронна мережа. При виборі кількості нейронів слід керуватися наступним правилом: *«кількість зв'язків між нейронами має бути значно менше кількості прикладів в навчальній множині»*. Кількість зв'язків розраховується як зв'язок кожного нейрона зі всіма нейронами сусідніх шарів, включаючи зв'язки на вхідному і вихідному шарах. Дуже велика кількість нейронів може привести до так званого «перенавчання» мережі, коли вона видає добрі результати на прикладах, що входять в навчальну вибірку, але практично не працює на інших прикладах.

Навчання нейромережі. Після налаштування конфігурації мережі слід вибрати алгоритм її навчання.

Метод зворотного поширення помилки – ітеративний градієнтний алгоритм навчання, який використовується з метою мінімізації середньоквадратичного відхилення поточних значень виходів мережі від потрібних. Однією з найважливіших властивостей алгоритму зворотного поширення помилки є *висока стійкість*, а отже, *надійність*. Хоча нейронні мережі, що використовують алгоритм зворотного поширення помилки, будучи потужним інструментом пошуку закономірностей, прогнозування і якісного аналізу, набули широкого поширення, їм властиві деякі недоліки. До них відноситься невисока швидкість збіжності (велике число необхідних ітерацій), що робить процес навчання дуже довгим і непридатним використання даного алгоритму для широкого круга задач, які вимагають швидкого рішення. Інші

алгоритми навчання НМ, хоча і працюють швидше, в більшості випадків, володіють меншою стійкістю.

Для алгоритму зворотного поширення помилки потрібно вказати два параметри:

- швидкість навчання - визначає величину кроку при ітераційній корекції вагів в нейронній мережі (рекомендується в інтервалі $0...1$). При великій величині кроку, збіжність буде швидшою, але є небезпека «перестрибнути» через рішення. З іншого боку, при малій величині кроку, навчання зажадає дуже багатьох ітерацій. На практиці величина кроку береться пропорційній крутості схилу, так, що алгоритм сповільнюється поблизу мінімуму. Правильний вибір швидкості навчання залежить від конкретної задачі і зазвичай шукається дослідним шляхом;

- момент - задається в інтервалі $0...1$. Рекомендується значення 0.9 ± 0.1 .

Method Resilient Propagation (Rprop) - еластичне поширення. Алгоритм використовує так зване «навчання по епохах», коли корекція вагів відбувається після пред'явлення мережі всіх прикладів з навчальної вибірки. Перевага даного методу полягає в тому, що він забезпечує збіжність, а, отже, і навчання мережі в 4-5 разів швидше, ніж алгоритм зворотного поширення. Для алгоритму Resilient Propagation вказуються параметри:

- крок спуску - коефіцієнт збільшення швидкості навчання, який визначає крок збільшення швидкості навчання при недосягненні алгоритмом оптимального результату;

- крок підйому - коефіцієнт зменшення швидкості навчання. Задається крок зменшення швидкості навчання в разі пропуску алгоритмом оптимального результату.

Далі необхідно задати умови, при виконанні якого навчання буде припинено:

- вважати приклад розпізнаним, якщо помилка менша - критерієм останову в даному випадку є умова, що розузгодження між еталонним і реальним виходом мережі стає менше заданого значення;

- після досягнення епохи - установка даного режиму дозволяє задати число епох (циклів навчання) після досягнення якого навчання зупиняється незалежно від величини помилки. Якщо прапорець скинутий, то навчання продовжуватиметься, поки помилка не стане менше заданого значення, але при цьому є вірогідність зациклення, коли помилка ніколи не буде досягнута. Тому, все-таки, бажано встановити прапорець по «Достижению эпохи», аби алгоритм був зупинений в будь-якому разі.

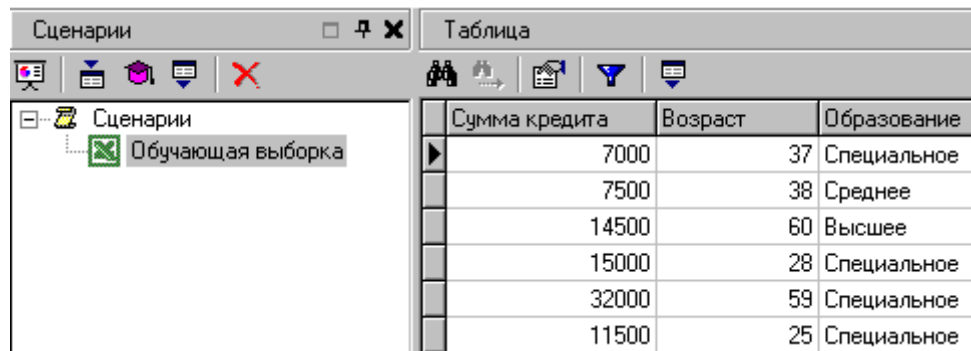
Тепер все готово до процесу навчання мережі. На початку всі ваги нейромережі ініціалізувалися випадковими значеннями. Після навчання ці ваги набувають певних значень. Навчання може з великою часткою вірогідності вважатися успішним, якщо відсоток розпізнаних прикладів на навчальній і тестовій множині досить великий (близький до 100%).

Порядок виконання роботи.

1. Створення навчальної вибірки. Розглянемо приклад побудови системи оцінки кредитоспроможності фізичної особи. Передбачимо, що експерти визначили основні чинники, що визначають кредитоспроможність. Ними виявилися: вік, освіта, площа квартири, наявність автомобіля, тривалість мешкання в даному регіоні. В організації була накопичена статистика повернень або не повернень взятих кредитів. Ця статистика представлена таблицею в MS Excel.

Сумма кредита	Возраст	Образование	Площадь квартиры	Автомобиль	Срок проживания	Давать кредит
7000	37	Специальное	37	отечественная	22	Да
7500	38	Среднее	29	импортная	12	Да
14500	60	Высшее	34	Нет автомобиля	30	Нет
15000	28	Специальное	14	отечественная	21	Да
32000	59	Специальное	53	отечественная	29	Да
11500	25	Специальное	28	отечественная	9	Да
5000	57	Специальное	18	отечественная	34	Да
61500	29	Высшее	26	Нет автомобиля	18	Нет
13500	37	Специальное	46	отечественная	28	Нет
25000	36	Специальное	20	Нет автомобиля	21	Нет
25500	68	Высшее	45	отечественная	30	Нет
...	...	...	...	...	...	...

Використовуючи «Мастер импорта» завантажимо файл даних з MS Excel в Deductor (рис. 7.2).



Сценарии	Таблица
Сценарии	Сумма кредита
Обучающая выборка	Возраст
	Образование
	7000
	37
	Специальное
	7500
	38
	Среднее
	14500
	60
	Высшее
	15000
	28
	Специальное
	32000
	59
	Специальное
	11500
	25
	Специальное

Рис. 7.2. Завантаження даних.

2. Нормалізація полів. Поля «Сума кредиту», «Вік», «Площа квартири» і «Тривалість мешкання» – безперервні значення, які перетворимо до інтервалу $[-1...1]$. «Освіта» представлена трьома унікальними значеннями, які можна порівнювати на більше або менше, а точніше краще або гірше. Тобто «Освіту» можна упорядкувати так: середнє, спеціальне, вище. Значення поля з наявністю автомобіля упорядкувати не можна. Його потрібно перетворити до бітової маски. Для кодування трьох значень потрібний два біта. Отже, це поле буде розбито на два (рис. 7.3).

Наличие автомобиля	Первый бит маски	Второй бит маски
Импортная	0	0
Отечественная	0	1
нет автомобиля	1	0

Рис. 7.3. Розбиття поля «Наявність автомобіля».

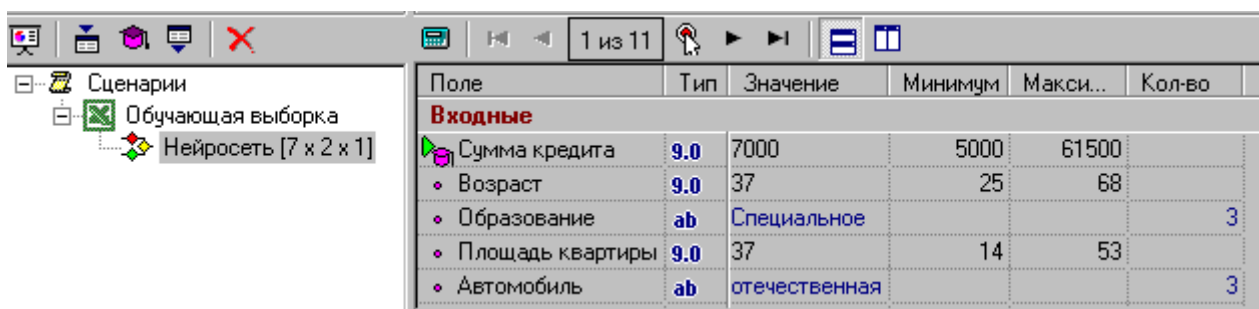
Скориставшись «Мастером обработки» (кнопка на панелі ) в 1-му вікні вибираємо спосіб обробки – «Нейросеть». Потім виконуємо налаштування призначень стовпців і в цьому ж вікні, ставимо відмітку в полі – «Настройка нормализации в ручную». У 4-му вікні майстра проводимо налаштування нормалізації стовпців відповідно до приведених вище міркувань.

3. Навчальну вибірку розіб'ємо на навчальну і тестову множину так, як програма пропонує це зробити за умовчанням, тобто випадкові 95 відсотків записів будуть в навчальній безлічі, останні 5 відсотків – в тестовому. (Крок 4 – Майстер обробки).

4. Конфігурація мережі. У вхідному шарі – 7 нейронів, тобто по одному нейрону на один вхід (у навчальній вибірці 6 стовпців, але стовпець «Автомобіль» представлений бітовою маскою з двох біт, для кожного з яких створений новий вхід). Зробимо один прихований шар з двома нейронами. У вихідному шарі буде один нейрон, на виході якого буде рішення про видачу кредиту (Крок 5 – Майстер обробки).

5. Виберемо алгоритм навчання мережі – Resilient Propagation з налаштуваннями за умовчанням. Умову закінчення навчання залишимо без зміни (Крок 6, 7, 8 – Майстер обробки).

6. Навчену таким чином нейромережу можна використовувати для ухвалення рішення про видачу кредиту фізичній особі. Це можна зробити, використовуючи аналіз «Що-якщо». Для його включення потрібно вибрати візуалізацію «Що-якщо». Тоді відкриється форма представлена на рис. 7.4 (Крок 9, 10 – Майстер обробки).



Поле	Тип	Значение	Минимум	Макси...	Кол-во
Входные					
Сумма кредита	9.0	7000	5000	61500	
Возраст	9.0	37	25	68	
Образование	ab	Специальное			3
Площадь квартиры	9.0	37	14	53	
Автомобиль	ab	отечественная			3

Рис. 7.4. Застосування візуалізатора «Що-якщо».

Після зміни в цій таблиці вхідних полів система сама приймає рішення про видачу кредиту і в полі «Давати кредит» проставляє або «Так», або «Ні». Стовпці «Мінімум» і «Максимум» визначають діапазон значень, на яких навчалася нейромережа. Слід дотримуватися цих обмежень, хоча і можливо взяти значення що трохи виходять за кордони діапазону.

7. Окрім такої таблиці аналіз «Що-якщо» містить діаграму, на якій відображується залежність вихідного поля від одного з вхідних полів при фіксованих значеннях останніх полів. Наприклад, потрібно взнати, на яку суму кредиту може розраховувати чоловік що володіє певними характеристиками. Це можна визначити по діаграмі (рис. 7.5).

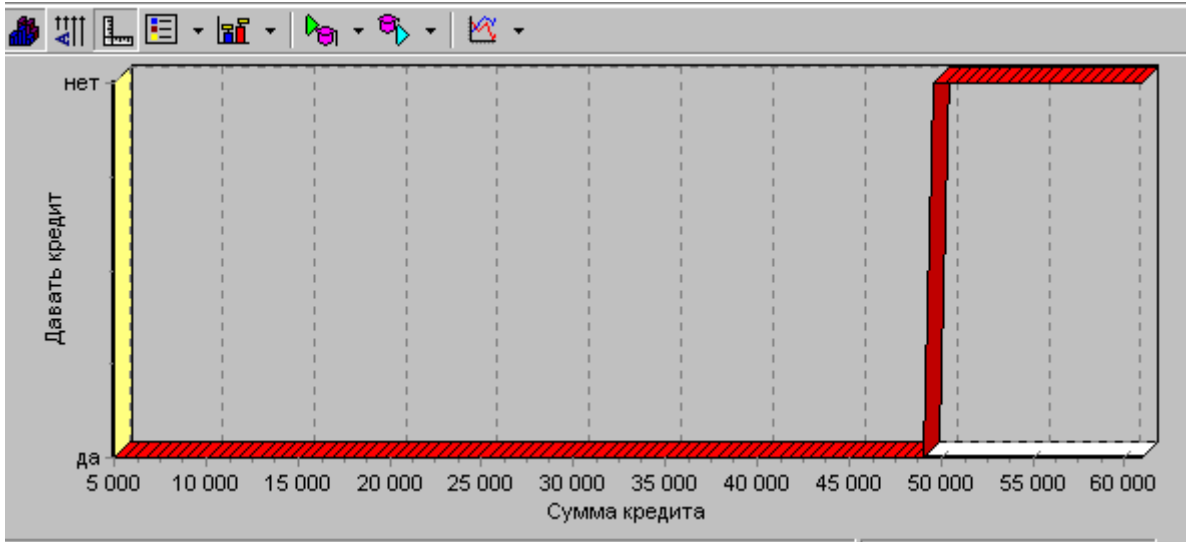


Рис. 7.5. Діаграма результатів аналізу.