Лекція 10. Класифікація за допомогою штучних нейронних мереж [1, 2]

Нейромережеві моделі розпізнавання

Нейронні мережі є спробою моделювання здатності людського мислення, зокрема, здатності навчатися і вирішувати задачі розпізнавання по прецедентах. Вони засновані на досягненнях біології і медицини — найпростіших моделях людського мозку, створених у середині минулого століття. Біологічний мозок розглядається як множина елементарних елементів — нейронів, з'єднаних один з одним численними зв'язками. Нейрони бувають трьох типів: рецептори (приймаючі сигнали з зовнішнього середовища і передавальні іншим нейронам), внутрішні нейрони (приймаючі сигнали від інших нейронів, що перетворюють їх і передають іншим нейронам) і реагуючі нейрони (приймаючі сигнали від нейронів і сигнали, що виробляють, у зовнішнє середовище).

Розглянемо найпростішу і найбільш розповсюджену модель людського мозку, орієнтовану на рішення задач розпізнавання – штучну багатошарову нейронну мережу.

Елементарним осередком нейронної мережі є модель штучного нейрона (рис. 10.1).

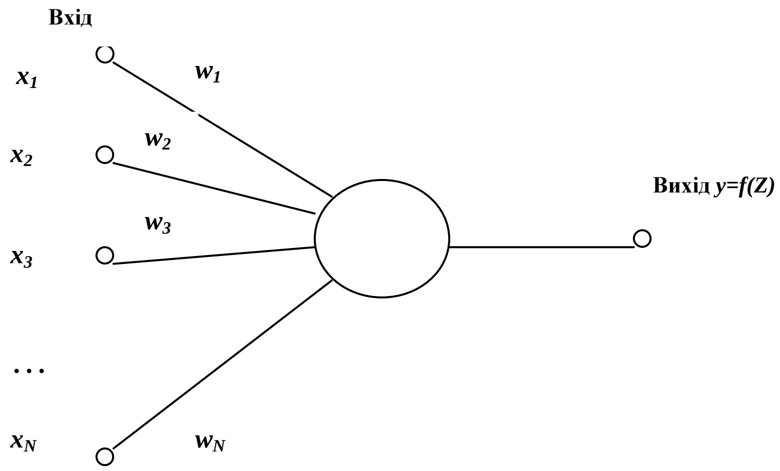

Рис. 10.1. Модель штучного нейрона

Кожен внутрішній чи реагуючий нейрон має множину вхідних зв'язків (синапсів), по яких надходять сигнали від інших внутрішніх нейронів чи рецепторів, і один вихідний зв'язок (аксон). Кожен зв'язок має деякий “вагу” w_i. При надходженні на вхід нейрона сукупності сигналів $x_1, x_2, \dots, x_N$ вони “підсилюються” з відповідними весами $w_1, w_2, \dots, w_N$. Нейрон переходить у стан, числова оцінка якого обчислюється як $Z = \sum_{i=1}^{N} w_i x_i$. Величина вихідного сигналу обчислюється як $f(Z)$, де f - активаційна функція. Приклади активаційних функцій приведені на рис. 10.2-10.5.

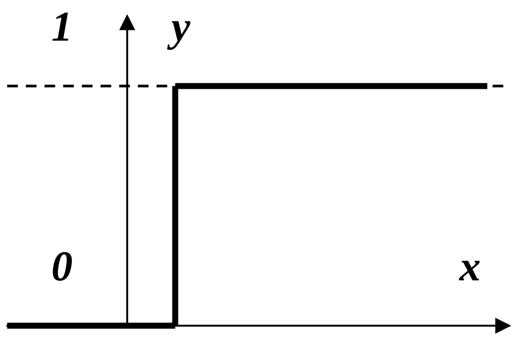

Рис. 10.2. Функція одиничного стрибка

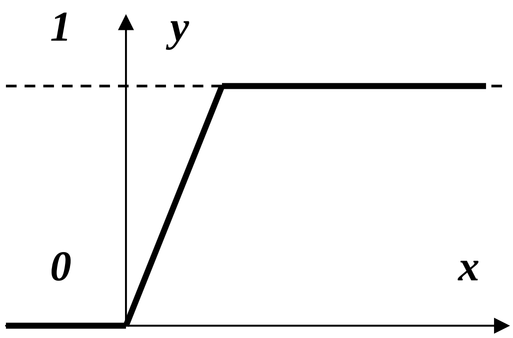

Рис. 10.3. Одиничний стрибок із лінійним порогом

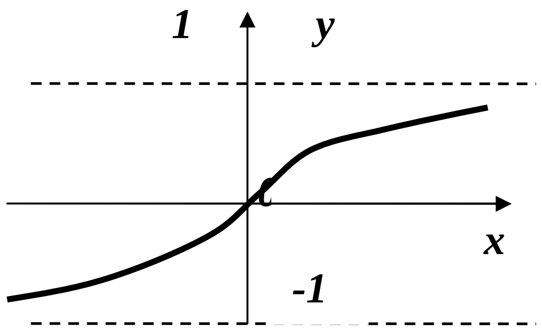

Рис. 10.4. Гіперболічний тангенс *f(x)=th(x)*

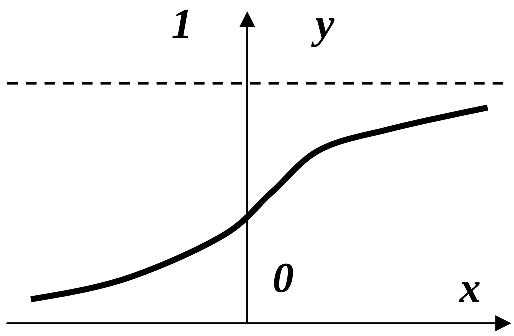

Рис. 10.5. Функція сигмоїда *f(x)=1/(1+exp(-ax))*

Нейрон вважається “збудженим”, якщо вихідний сигнал відмінний від нуля, а величина $y = f(\sum_{i=1}^{N} w_i x_i)$ характеризує ступінь збудження. Вид функцій і область їх зміни відбивають апріорні представлення про функціонування біологічних нейронів: величина збудження залежить монотонно від стану, обмежена знизу і зверху, і “сильно” змінюється в невеликому інтервалі значень $Z = \sum_{i=1}^{N} w_i x_i$.

Найбільш простими, розповсюдженими і дослідженими є багатошарові нейронні мережі прямої дії. Загальний вид подібної мережі зображений на рис. 10.6. Мережа складається з N шарів, кожні шар складається з n_i нейронів, кожен нейрон *j*-го рівня зв'язаний з кожним нейроном *j*+1 – го рівня. Фіктивний нульовий шар складається з *n* вхідних нейронів, на кожний з який подається значення деякої ознаки x_i. Результатами класифікації є вихідні значення нейронів N -го шару.

Розпізнаваний об'єкт *S* надходить на 0-й шар. Далі сигнал, що надійшов (ознаковий опис) послідовно перетвориться по шарах згідно заданим фіксованим весам синаптичних зв'язків і обраної активаційної функції. Якщо $y_j^N(S), j = 1,2,...,l,$ є значенням *j-го* нейрона вихідного шару, то інформаційний вектор результатів класифікації S обчислюється згідно (10.1).

$$\alpha_j^A(S) = \begin{cases} 1, & y_j^N(S) > 0, \\ \Delta, & y_j^N(S) = 0, \\ 0, & y_j^N(S) < 0. \end{cases} \tag{10.1}$$

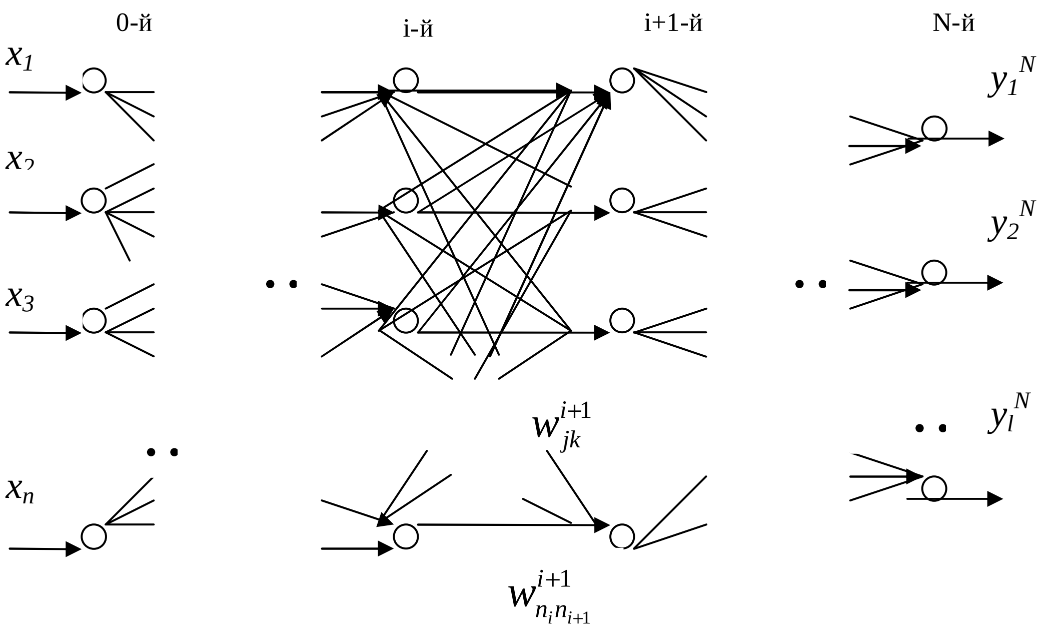

Рис. 10.7. Структурна схема нейронної класифікуючої мережі прямої дії

Значення невідомих вагових коефіцієнтів знаходяться в результаті процесу навчання мережі. Початкові значення вагових коефіцієнтів задаються випадково. Об'єкти навчання послідовно надходять на вхід мережі. Якщо пред'явлений об'єкт класифікується правильно, коефіцієнти залишаються колишніми і на вхід мережі надходить наступний об'єкт. Якщо при класифікації об'єкта відбувається помилка, вагові коефіцієнти змінюються певним чином. Для одношарової мережі вони змінюються згідно з простою ітераційною формулою подібно використовуваній в методі потенційній функцій. Для багатошарової мережі використовуються спеціальні рекурентні формули перерахування вагових коефіцієнтів від останнього рівня до першого (метод "зворотного поширення помилки"). Навчання закінчується, якщо зміна коефіцієнтів не приводить до подальшого зменшення сумарного числа помилок на навчальній вибірці.

Перцептронне представлення

Основною теоремою в теорії перцептронів є теорема Розенблатта про навчання перцептрону, яка стверджує, що перцептрон здатний навчитися всьому, що можна подати у вигляді перцептронної функції. Важливо розрізнювати *представимість* і *здатність до навчання*. Поняття представимості означає здатність перцептрону моделювати певну функцію. Здатність до навчання вимагає наявності систематичної процедури налаштування ваг мережі для реалізації цієї функції.

Проблема функції ВИКЛЮЧАЮЧЕ АБО

Теорема Мінського стверджує, що одношаровий перцептрон не може відтворити навіть таку просту функцію, як ВИКЛЮЧАЮЧЕ АБО. Це функція від двох аргументів, кожний з яких може бути нулем або одиницею. Вона приймає значення одиниці, коли один з аргументів рівний одиниці (але не обидва). Проблему можна проілюструвати за допомогою одношарової однонейронної системи з двома входами, показаною на рис. 10.8.

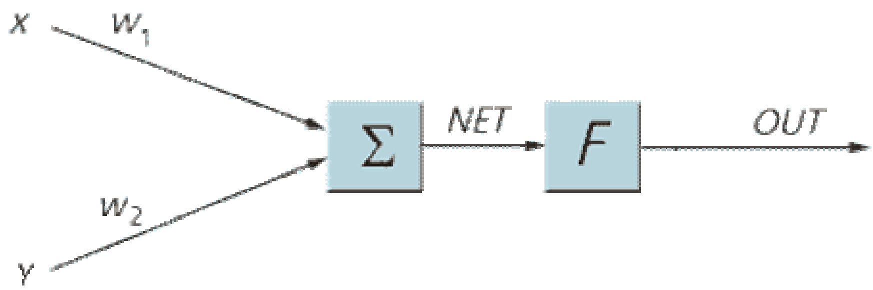

Рис. 10.8. Однонейронна система

Визначимо один вхід через *x*, а інший через *y*, тоді всі їх можливі комбінації будуть складатися з чотирьох точок на площині x-y, як показано на рис.10.9. Наприклад, точка x=0 і y=0 позначена на рисунку як точка A_0.

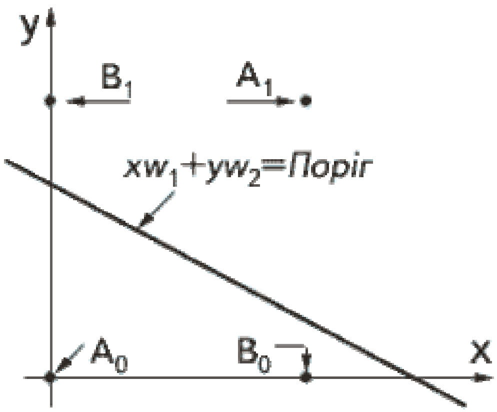

Рис. 10.9. Проблема ВИКЛЮЧАЮЧЕ АБО

Таблиця 10.1 показує необхідний зв'язок між входами і виходом, де вхідні комбінації, які повинні давати нульовий вихід, помічені A_0 і A_1, одиничний вихід - B_0 і B_1.

Таблиця 10.1. Таблиця істинності для функції ВИКЛЮЧАЮЧЕ АБО

Точки	Значення *x*	Значення *y*	Необхідний вихід
A_0	0	0	0
B_0	1	0	1
B_1	0	1	1
A_1	1	1	0

У мережі на рис. 10.8 функція *F* є звичайним порогом, так що OUT приймає значення нуль, коли NET менше 0,5, і одиницю у випадку, коли *NET* більше або дорівнює 0,5. Нейрон виконує наступне обчислення:

$$NET = xw_1 + yw_2 \qquad (10.2)$$

Ніяка комбінація значень двох ваг не може дати співвідношення між входом і виходом, що задається у таблиці 10.1. Щоб зрозуміти це обмеження, зафіксуємо NET на величині порога 0,5. Мережа в цьому випадку описується рівнянням (10.2). Це рівняння лінійно по *x* і *y*, тобто всі значення по *x* і *y*, що задовольняють цьому рівнянню, будуть лежати на деякій прямій в площині *x*-*y*.

$$xw_1 + yw_2 = 0{,}5 \qquad (10.3)$$

Будь-які вхідні значення для *x* і *y* на цій лінії будуть давати порогове значення 0,5 для *NET*. Вхідні значення з одного боку прямої забезпечать значення *NET* більше порога, отже, *OUT*=1. Вхідні значення по іншу сторону прямої забезпечать значення *NET* менше порогового значення, роблячи *OUT* рівним 0. Зміни значень w_1, w_2 і порога будуть міняти нахил і розташування прямої. Для того щоб мережа реалізовувала функцію ВИКЛЮЧАЮЧЕ АБО, задану у таблиці 10.2, треба розташувати пряму так, щоб точки А були з одного боку прямої, а точки В — з іншою. Спробувавши намалювати таку пряму на рис. 10.9, переконуємося, що це неможливе. Це означає, що які б значення ні приписувалися вагам і порогу, мережа нездібна відтворити співвідношення між входом і виходом, необхідне для представлення функції що ВИКЛЮЧАЮЧЕ АБО.

Поглянувши на задачу з іншої точки зору, розглянемо *NET* як поверхню над площиною *x*-*y*. Кожна точка цієї поверхні знаходиться над відповідною точкою площини *x*-*y* на відстані, рівній значенню *NET* в цій точці. Можна показати, що нахил цієї *NET*-поверхні однаковий для всієї поверхні *x*-*y*. Всі точки, в яких значення *NET* дорівнює величині порога, проектуються на лінію рівня площини *NET* (див. рис. 10.10).

Рис. 10.10. Персептронна *NET*-площина

Ясно, що всі точки по один бік порогової прямої проектуються в значення *NET*, більше за поріг, а точки по інший бік дадуть менші значення *NET*. Таким чином, порогова пряма розбиває площину *x*-*y* на дві області. У всіх точках по один бік порогової прямої значення *OUT* дорівнює одиниці, по інший бік - нулю.

Лінійна роздільність

Як ми бачили, неможливо намалювати пряму лінію, що розділяє площину *x*-*y* так, щоб реалізовувалася функція ВИКЛЮЧАЮЧЕ АБО. На жаль, цей приклад не єдиний. Є широкий клас функцій, що не реалізовуються одношаровою мережею. Про ці функції кажуть, що вони є лінійно нероздільними, і вони накладають певні обмеження на можливості одношарові мереж.

Лінійна роздільність обмежує одношарові мережі задачами класифікації, в яких множини точок (відповідних вхідним значенням) можуть бути розділені геометрично. Для нашого випадку з двома входами роздільник є прямою лінією. У разі трьох входів розділення здійснюється площиною, що розтинає тривимірний простір. Для чотирьох або більше входів візуалізація неможлива і необхідно уявно представити *n*-мірний простір, що розтинається "гіперплощиною" геометричним об'єктом, який розтинає простір чотири або більшого числа вимірів.

Оскільки лінійна роздільність обмежує можливості перцептронного представлення, то важливо знати, чи є дана функція роздільної. На жаль, не існує простого способу визначити це, якщо число змінних велике.

Нейрон з *n* двійковими входами може мати 2^n різних вхідних образів, що складаються з нулів і одиниць. Оскільки кожний вхідний образ може відповідати двом різним бінарним виходам (одиниця і нуль), то всього є 2^{2n} функцій від *n* змінних.

Таблиця 10.2. Лінійно роздільні функції

n	2^{2n}	Число лінійно роздільних функцій
1	4	4
2	16	14
3	256	104
4	65536	1882
5	4,3x109	94572
6	1,8x1019	15 028 134

(Взято з R. 0. Winder, Single-stage logic. Paper presented at the AIEE Fall General Meeting, 1960.)

Як видно з таблиці 10.2, імовірність того, що випадково вибрана функція виявиться лінійно роздільною, досить мала навіть для помірного числа змінних. З цієї причини одношарові перцептрони на практиці обмежені простими задачами.

Подолання обмеження лінійної роздільності

До кінця 60-х років проблема лінійної роздільності була добре зрозуміла. До того ж було відомо, що це серйозне обмеження представлення одношаровими мережами можна подолати, додавши додаткові прошарки. Наприклад, двошарові мережі можна отримати каскадним з'єднанням двох одношарових мереж. Вони здатні виконувати більш загальні класифікації, відділяючи ті точки, які містяться в опуклих обмежених або необмежених областях. Область називається опуклою, якщо для будь-яких двох її точок, що з'єднують їх відрізок повністю лежить в області. Область називається обмеженою, якщо її можна взяти в деяке коло. Необмежену область неможливо укласти всередину кола (наприклад, область між двома паралельними лініями). Приклади опуклих обмежених і необмежених областей наведені на рис. 10.11.

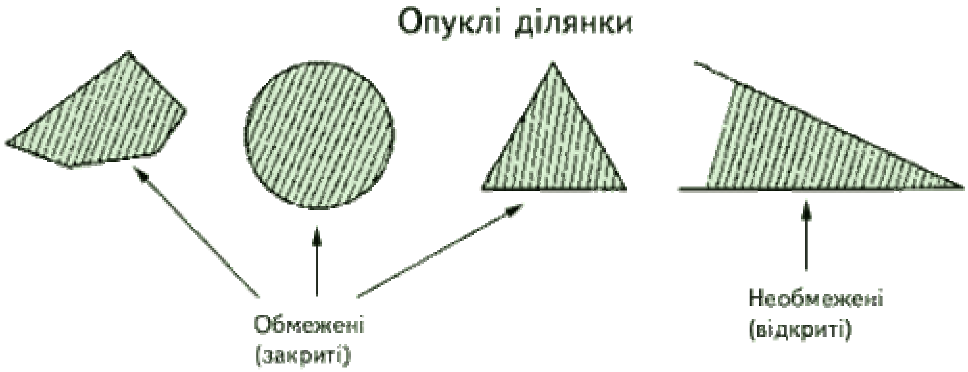

Рис. 10.11. Опуклі обмежені і необмежені області

Щоб уточнити вимогу опуклості, розглянемо просту двошарову мережу з двома входами, підведеними до двох нейронів першого прошарку, сполученими з єдиним нейроном в шарі 2 (див. рис. 10.12). Нехай поріг вихідного нейрона рівний 0,75, а обидва його ваги рівні 0,5. У цьому випадку для того, щоб поріг був перевищений і на виході з'явилася одиниця, потрібно, щоб обидва нейрони першого рівня на виході мали одиницю. Таким чином, вихідний нейрон реалізовує логічну функцію І. На рис. 10.12 кожний нейрон прошарку 1 розбиває площину *x-y* на дві півплощини, один забезпечує одиничний вихід для входів нижче верхньої лінії, інший - для входів вище нижньої лінії. На рис. 10.12 показаний результат такого подвійного розділення, де вихідний сигнал нейрона другого прошарку рівний одиниці тільки всередині V-образної області. Аналогічно у другому прошарку може бути використано три нейрони з подальшим розділенням площини і створенням області трикутної форми. Включенням достатнього числа нейронів у вхідний прошарок може бути утворений опуклий багатокутник будь-якої бажаної форми. Оскільки вони утворені за допомогою операції І над областями, що задаються лініями, то всі такі многогранники опуклі, отже, тільки опуклі області і виникають. Точки, що не складають опуклої області, не можуть бути відділені від інших точок площини двошаровою мережею.

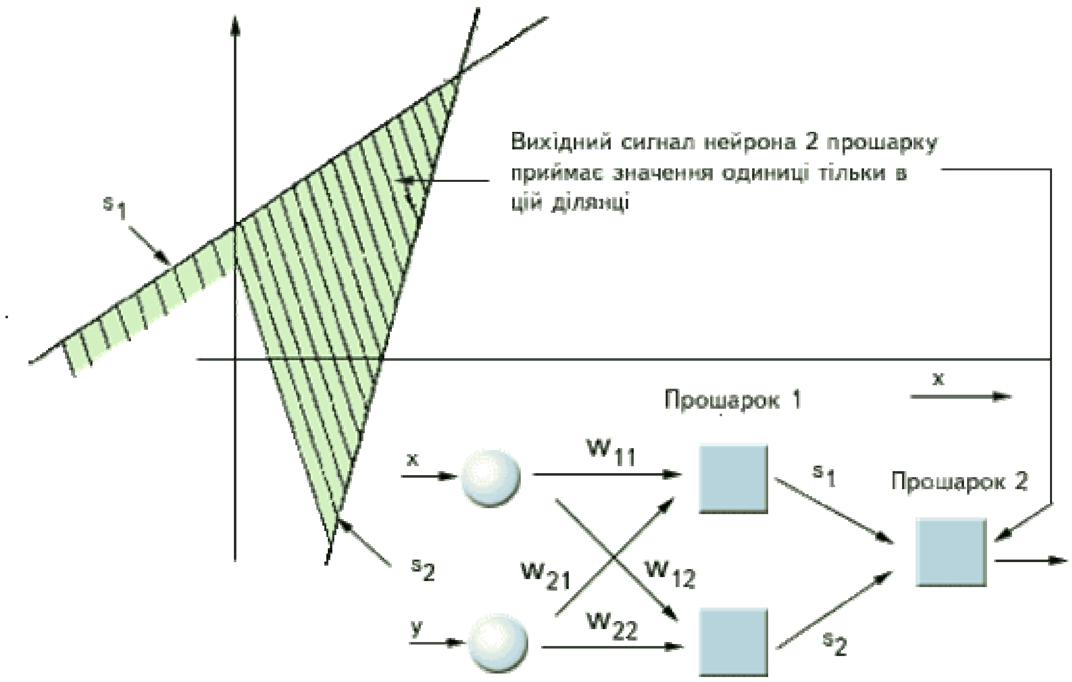

Рис. 10.12. Опукла область рішень, що задається двошаровою мережею

Нейрон другого прошарку не обмежений функцією I. Він може реалізовувати багато інших функцій при відповідному виборі ваг і порога. Наприклад, можна зробити так, щоб одиничний вихід будь-якого з нейронів першого прошарку приводив до появи одиниці на виході нейрона другого прошарку, реалізувавши тим самим логічне АБО. Є 16 двійкових функцій від двох змінних. Якщо вибирати відповідним образом ваги і поріг, то можна відтворити 14 з них (все, крім ВИКЛЮЧАЮЧЕ АБО і що ВИКЛЮЧАЮЧЕ НІ).

Входи не обов'язково повинні бути двійковими. Вектор безперервних входів може являти собою довільну точку на площині *x-y*. У цьому випадку ми маємо справу зі здатністю мережі розділяти площину на безперервні області, а не з розділенням дискретних множин точок. Для всіх цих функцій лінійна роздільність показує, що вихід нейрона другого прошарку рівний одиниці тільки в частині площини *x-y*, обмеженої багатокутною областю. Тому для розділення площин *P* і *Q* необхідно, щоб все *P* лежали всередині опуклої багатокутної області, що не містить точок *Q* (або навпаки).

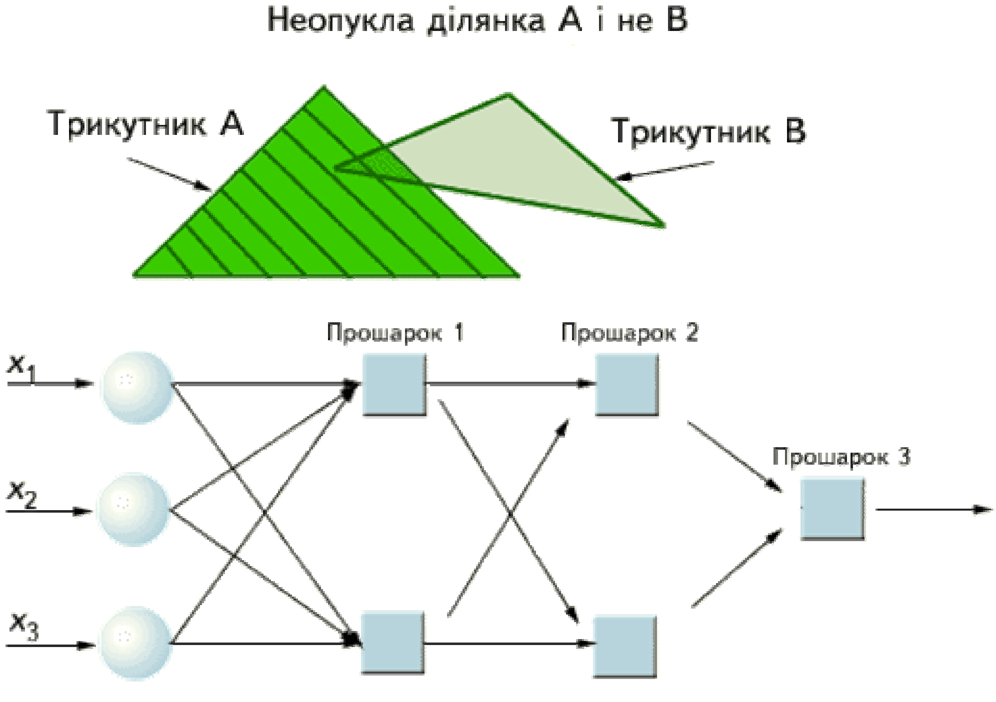

Рис. 10.13. "Неопукла" область рішень, що задається тришаровою мережею

Тришарова мережа є більш загальної. Її класифікуючі можливості обмежені лише числом штучних нейронів і ваг. Обмеження на опуклість відсутні. Тепер нейрон третього прошарку приймає як вхід набір опуклих багатокутників, і їх логічна комбінація може бути неопуклою. На рис. 10.13 ілюструється випадок, коли два трикутники *A* і *B*, скомбіновані за допомогою функцій "*A* і не *B*", задають неопуклу область. При додаванні нейронів і ваг число сторін багатокутників може необмежено зростати. Це дозволяє апроксимувати область будь-якої форми з будь-якою точністю. Додатково не всі вихідні області другого прошарку повинні перетинатися. Можливо об'єднувати різні області, опуклі і неопуклі, видаючи на виході одиницю кожний раз, коли вхідний вектор належить одній з них.

Незважаючи на те що можливості багатошарових мереж були відомі давно, протягом багатьох років не було теоретично обґрунтованого алгоритму для налаштування їх ваг. В подальших розділах ми детально вивчимо багатошарові навчальні алгоритми, але зараз досить розуміти проблему і знати, що дослідження привели до певних результатів.

Ефективність запам'ятовування

Серйозні питання є відносно ефективності запам'ятовування інформації в перцептроні (або будь-яких інших нейронних мережах) в порівнянні із звичайною комп'ютерною пам'яттю і методами пошуку інформації в ній. Наприклад, в комп'ютерній пам'яті можна зберігати всі вхідні образи разом з класифікуючими бітами. Комп'ютер повинен знайти необхідний образ і дати його класифікацію. Для прискорення пошуку могли б бути використані різні добре відомі методи. Якщо точна відповідність не знайдена, то для відповіді може бути використане правило найближчого сусіда.

Число бітів, необхідне для зберігання цієї ж інформації у вагах перцептрону, може бути значно меншим в порівнянні з методом звичайної комп'ютерної пам'яті, якщо образи допускають економний запис. Однак Мінський розробив патологічні приклади, в яких число бітів, необхідних для представлення ваг, росте з розмірністю задачі швидше, ніж експонентне. У цих випадках вимоги до пам'яті із зростанням розмірності задачі швидко стають нездійсненними. Якщо, як він передбачив, ця ситуація не є виключенням, то перцептрони часто можуть бути обмежені тільки малими задачами. Наскільки загальними є такі неподатливі множини образів? Це залишається відкритим питанням, що відноситься до всіх нейронних мереж. Пошуки відповіді надзвичайно важливі для досліджень по нейронних мережах.

Навчання перцептрону

Здатність штучних нейронних мереж навчатися є їх найбільш інтригуючою властивістю. Подібно до біологічних систем, які вони моделюють, нейронні мережі самі моделюють себе внаслідок спроб досягнути кращої моделі поведінки.

Використовуючи критерій лінійної роздільності, можна вирішити, чи здатна одношарова нейронна мережа реалізовувати необхідну функцію. Навіть в тому випадку, коли відповідь позитивна, це принесе малу користь, якщо у нас немає способу знайти потрібні значення для ваг і порогів. Щоб мережа мала практичну цінність, потрібен систематичний метод (алгоритм) для обчислення цих значень. Розенблат зробив це в своєму алгоритмі навчання перцептрону разом з доказом того, що перцептрон може бути навчений всьому, що він може реалізовувати.

Навчання може бути з вчителем або без нього. Для навчання з вчителем потрібен "зовнішній" вчитель, який оцінював би поведінку системи і керував її подальшими модифікаціями. При навчанні без вчителя, що розглядається в подальших розділах, мережа шляхом самоорганізації робить необхідні зміни. Навчання перцептрону є навчанням з вчителем.

Алгоритм навчання перцептрону може бути реалізований на цифровому комп'ютері або іншому електронному пристрої, і мережа стає в певному значенні самоорганізуючою. З цієї причини процедуру налаштування ваг переважно називають "навчанням" і кажуть, що мережа "навчається". Доказ Розенблата став основною віхою і дав могутній імпульс дослідженням в цій області. Сьогодні в тій або іншій формі елементи алгоритму навчання перцептрону зустрічаються в багатьох мережевих парадигмах.

Алгоритм навчання перцептрону

Перцептрон навчають, подаючи множину образів по одному на його вхід і налаштовуючи ваги доти, поки для всіх образів не буде досягнуть необхідний вихід. Допустимо, що вхідні образи нанесені на демонстраційні карти. Кожна карта розбита на квадрати і від кожного квадрата на перцептрон подається вхід. Якщо в квадраті є лінія, то від нього подається одиниця, в іншому випадку - нуль. Множина квадратів на карті задає, таким чином, множину нулів і одиниць, яка і подається на входи перцептрону. Мета полягає в тому, щоб навчити перцептрон включати індикатор при подачі на нього множини входів, що задають непарне число, і не включати у разі парного.

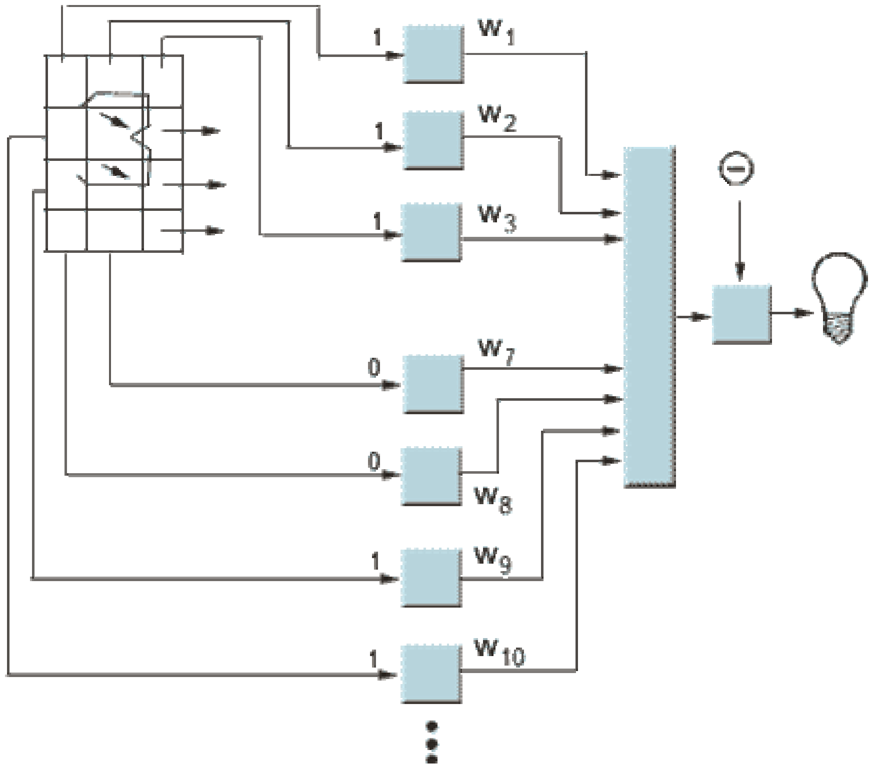

Рис. 10.14. Перцептронна система розпізнавання зображень

На рис. 10.14 показана перцептронная конфігурація. Припустимо, що вектор X є образом розпізнаваної демонстраційної карти. Кожна компонента (квадрат) X - (x_1, x_2, ..., x_n) множиться на відповідну компоненту вектора ваг W - (w_1, w_2,..., w_n). Ці добутки підсумовуються. Якщо сума перевищує поріг Θ, тоді вихід нейрона Y рівний одиниці (індикатор запалюється), в іншому випадку він - нуль. Як ми бачили в розділі 1, ця операція компактно записується у векторній формі як Y = XW, а після неї слідує порогова операція.

Для навчання мережі образ X подається на вхід і обчислюється вихід Y. Якщо Y правильний, то нічого не міняється. Однак якщо вихід неправильний, то ваги, приєднані до входів, що підсилюють помилковий результат, модифікуються, щоб зменшити похибку.

Щоб побачити, як це здійснюється, припустимо, що демонстраційна карта з цифрою 3 подана на вхід і вихід Y рівний 1 (показуючи непарність). Оскільки це правильна відповідь, то ваги не змінюються. Якщо, однак, на вхід подається карта з номером 4 і вихід Y рівний одиниці (непарний), то ваги, приєднані до одиничних входів, повинні бути зменшені, оскільки вони прагнуть дати невірний результат. Аналогічно, якщо карта з номером 3 дає нульовий вихід, то ваги, приєднані до одиничних входів, повинні бути збільшені, щоб скоректувати похибку.

Цей метод навчання може бути підсумований таким чином:

1. Подати вхідний образ і обчислити Y.
2. Якщо вихід правильний, то перейти на крок 1;
3. Якщо вихід неправильний і рівний нулю, то додати всі входи до відповідної ним ваги;або
4. Якщо вихід неправильний і рівний одиниці, то відняти кожний вхід з відповідної йому ваги.
5. Перейти на крок 1.

За кінцеве число кроків мережа навчиться розділяти карти на парні і непарні при умові, що множина цифр лінійно роздільна. Це означає, що для всіх непарних карт вихід буде більшим за поріг, а для всіх парних менше. Зазначимо, що це навчання глобальне, тобто мережа навчається на всій множині карт. Виникає питання про те, як ця множина повинна пред'являтися, щоб мінімізувати час навчання. Чи повинні елементи множини пред'являтися послідовно один за одним або карти потрібно вибирати випадково? Нескладна теорія служить тут путівником.

Дельта-правило

Важливе узагальнення алгоритму навчання перцептрону, зване дельтою-правилом, переносить цей метод на безперервні входи і виходи. Щоб зрозуміти, як воно було отримане, кроки 2-4 алгоритму навчання перцептрону можуть бути сформульовані в узагальненій формі за допомогою введення величини d, яка рівна різниці між необхідним або цільовим виходом T і реальним виходом Y

$$d = T - Y. \quad (10.4)$$

Випадок, коли d =0, відповідає кроку 2, коли вихід правильний і в мережі нічого не змінюється. Крок 3 відповідає випадку d> 0, а крок 4 випадку d< 0.

У будь-якому з цих випадків перцептронний алгоритм навчання зберігається, якщо d множиться на величину кожного входу x_i і цей добуток додається до відповідної ваги. З метою узагальнення вводиться коефіцієнт "швидкості навчання" h), який множиться на dx_i, що дозволяє управляти середньою величиною зміни ваги.

У алгебраїчній формі запису

$$D_i = hdx_i, \qquad (10.5)$$

$$w(n+1) = w(n) + D_i, \qquad (10.6)$$

де D_i - корекція, пов'язана з i-м входом x_i; $w_i(n+1)$ - значення ваги i після корекції; $w_i(n)$ - значення ваги i до корекції.

Дельта-правило модифікує ваги відповідно до необхідного і дійсного значень виходу кожної полярності як для безперервних, так і для бінарних входів і виходів. Ці властивості відкрили множину нових застосувань.

Труднощі з алгоритмом навчання перцептрону

Може виявитися скрутним визначити, чи виконана умова роздільності для конкретної навчальної множини. Крім того, в багатьох ситуаціях, що зустрічаються на практиці входи часто міняються у часі і можуть бути роздільні в один момент часу і нероздільні в іншій. У доказі алгоритму навчання персептрону нічого не говориться також про те, скільки кроків потрібно для навчання мережі. Мало втішливого в знанні того, що навчання закінчиться за кінцеве число кроків, якщо необхідний для цього час порівнянний з геологічною епохою. Крім того, не доведено, що перцептронний алгоритм навчання більш швидкий в порівнянні з простим перебором всіх можливих значень ваг, і в деяких випадках цей примітивний підхід може виявитися кращим.

На ці питання ніколи не знаходилося задовільної відповіді, вони відносяться до природи навчального матеріалу. Відповіді для сучасних мереж як правило не більш задовільні, ніж для перцептрону. Ці проблеми є важливою областю сучасних досліджень.

[1] Ф.Уосермен Нейрокомп'ютерна техніка: Теорія і практика. Переклад українською, І.Ю. Юрчак, 2001. http://www.victoria.lviv.ua/html/wosserman/index.htm

[2] Ю.И.Журавлев, В.В.Рязанов, О.В.Сенько Распознавание. Математические методы. Программная система. Практические применения. — М.: Фазис, 2005.