**ЛЕКЦІЯ 9. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ**

* [Біологічний прототип](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_1)
* [Штучний нейрон](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_2)
* [Одношарові штучні нейронні мережі](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_3)
* [Багатошарові штучні нейронні мережі](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_4)
* [Термінологія, позначення і схематичне зображення штучних нейронних мереж](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_5)
* [Висновки](http://www.victoria.lviv.ua/html/wosserman/rozdil1.htm#r1_6)

Штучні нейронні мережі надзвичайно різноманітні по своїх конфігураціях. Незважаючи на таку різноманітність, мережеві парадигми мають багато загального. У цьому розділі подібні питання розглядаються для того, щоб читач був знайомий з ними до того моменту, коли пізніше вони знов зустрінуться в книзі.

Використані позначення і графічні представлення тут були вибрані як найбільш широко використовувані в цей час (опублікованих стандартів не є), вони зберігаються протягом всієї книги.

**Біологічний прототип**

Розвиток штучних нейронних мереж надихається біологією. Тобто розглядаючи мережеві конфігурації і алгоритми, дослідники мислять в термінах організації мозкової діяльності. Але на цьому аналогія може і закінчитися. Наші знання про роботу мозку так обмежені, що мало б знайшлося керівних орієнтирів для тих, хто став би їм наслідувати. Тому розробникам мереж доводиться виходити за межі сучасних біологічних знань в пошуках структур, здатних виконувати корисні функції. У багатьох випадках це приводить до необхідності відмови від біологічної правдоподібності, мозок стає просто метафорою, і створюються мережі, неможливі в живій матерії або такі, що вимагають неправдоподібно великих допущень про анатомію і функціонування мозку.

Незважаючи на те що зв'язок з біологією слабкий і часто суттєвий, штучні нейронні мережі продовжують порівнюватися з мозком. Їх функціонування часто нагадує людське пізнання, тому важко уникнути цієї аналогії. На жаль, такі порівняння неплідні і створюють невиправдані очікування, неминуче ведучі до розчарування. Дослідницький ентузіазм, заснований на помилкових надіях, може випаруватися, зіткнувшись з суворою дійсністю, як це вже одного разу було в шістдесяті роки, і багатообіцяюча область знов прийде в занепад, якщо не буде дотримуватись необхідна стриманість.

Незважаючи на зроблені попередження, корисно все ж знати дещо про нервову систему ссавців, оскільки вона успішно вирішує задачі, до виконання яких лише прагнуть штучні системи. Подальше обговорення досить коротке. Додаток А містить більш широкий (але ні в якому разі не повний) розгляд нервової системи ссавців для тих, хто хоче дізнатися більше про цей чудовий предмет.

Нервова система людини, побудована з елементів, званих нейронами, має приголомшуючу складність. Біля 1011 нейронів беруть участь в приблизно 1015 передаючих зв'язках, що мають довжину більше за метр. Кожному нейрону притаманне багато властивостей, загальних з іншими елементами тіла, але його унікальною властивістю є прийом, обробка і передача електрохімічних сигналів по нервових шляхах, що утворюють комунікаційну систему мозку.

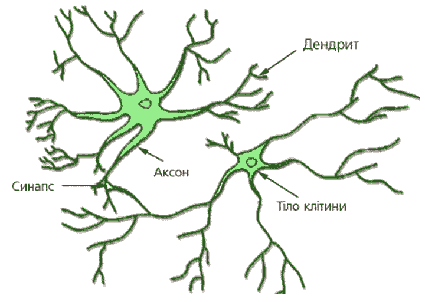


Рис. 1.1. Біологічний нейрон

На рис.1.1 показана структура пари типових біологічних нейронів. Дендрити йдуть від тіла нервової клітки до інших нейронів, де вони приймають сигнали в точках з'єднання, званих синапсами. Прийняті синапсом вхідні сигнали надходять до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть збудити нейрон, інші перешкодити його збудженню. Коли сумарне збудження в тілі нейрона перевищує деякий поріг, нейрон збуджується, посилаючи по аксону сигнал іншим нейронам. В цій основній функціональній схемі багато узагальнень і винятків, проте більшість штучних нейронних мереж моделюють лише ці прості властивості.

**Штучний нейрон**

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона поступає деяка множина сигналів, кожний з яких є виходом іншого нейрона. Кожний вхід перемножується з відповідною вагою, аналогічної синаптичній силі, і всі доданки підсумовуються, визначаючи рівень активації нейрона. На рис.1.2 представлена модель, що реалізує цю ідею. Хоча мережеві парадигми досить різноманітні, в основі майже всіх їх лежить ця конфігурація. Тут множина вхідних сигналів, позначених *x*1, *x*2*,..., xn*, надходить на штучний нейрон. Вхідні сигнали, що в сукупності позначені вектором *X*, відповідають сигналам, що надходять до синапсів біологічного нейрона. Кожний сигнал перемножується з відповідною вагою *w*1*, w*2*,..., wn,* і надходить на підсумовуючий блок, позначений S. Кожна вага відповідає "силі" одного біологічного синаптичного зв'язку. Множина ваг в сукупності позначається вектором *W*. Підсумовуючий блок, що відповідає тілу біологічного елемента, складає зважені входи алгебраїчно, створюючи вихід, який ми будемо називати *NET*. У векторних позначеннях це може бути компактно записане таким чином:

*NET = XW*.

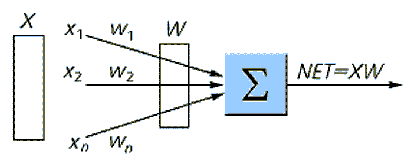


Рис. 1.2. Штучний нейрон

**Активаційна функція**

Сигнал *NET* далі, як правило, перетворюється активаційною функцією *F* і дає вихідний нейронний сигнал OUT. Активаційна функція може бути звичайною лінійною функцією

*OUT = K*(*NET*),

де *К* константа порогової функції

*OUT* = 1, якщо *NET* > *Т*,  
*OUT* = 0 в інших випадках,

де *Т* деяка постійна порогова величина, або ж функція, що точніше моделює нелінійну передатну характеристику біологічного нейрона і надає нейронній мережі великі можливості.

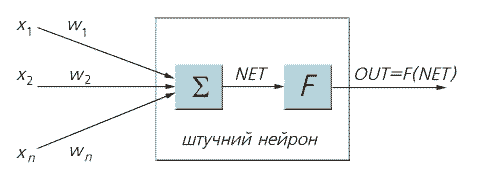


Рис.1.3. Штучний нейрон з активаційною функцією

На рис.1.3 блок, позначений *F*, приймає сигнал *NET* і видає сигнал *OUT*. Якщо блок *F* звужує діапазон зміни величини *NET* так, що при будь-яких значеннях *NET* значення *OUT* належать деякому кінцевому інтервалу, то *F* називається *"стискаючою" функцією*. В якості "стискаючої" функції часто використовується логістична або "сигмоїдальна" (*S*-образна) функція, показана на рис. 1.4а. Ця функція математично виражається як *F*(*х*) *=* 1*/(*1 + *е-х*). Таким чином,

http://www.victoria.lviv.ua/html/wosserman/images/rozdil1/Image40.gif.

Аналогічно з електронними системами активаційну функцію можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Коефіцієнт посилення обчислюється як відношення приросту величини OUT до невеликого приросту величини, що викликає NET. Він виражається нахилом кривої при певному рівні збудження і змінюється від малих значень при великих негативних збудженнях (крива майже горизонтальна) до максимального значення при нульовому збудженні і знов зменшується, коли збудження стає великим позитивним. Гросберг (1973) виявив, що подібна нелінійна характеристика вирішує поставлену ним дилему шумового насичення. Яким чином одна і та ж мережа може обробляти як слабкі, так і сильні сигнали? Слабі сигнали потребують великого мережевого посилення, щоб дати придатний до використання вихідний сигнал. Однак підсилювальні каскади з великими коефіцієнтами посилення можуть привести до насичення виходу шумами підсилювачів (випадковими флуктуаціями), які присутні в будь-якій фізично реалізованій мережі. Сильні вхідні сигнали в свою чергу також будуть приводити до насичення підсилювальних каскадів, виключаючи можливість корисного використання виходу. Центральна область логістичної функції, що має великий коефіцієнт посилення, вирішує проблему обробки слабих сигналів, в той час як області з падаючим посиленням на позитивному і негативному кінцях підходять для великих збуджень. Таким чином, нейрон функціонує з великим посиленням в широкому діапазоні рівня вхідного сигналу.

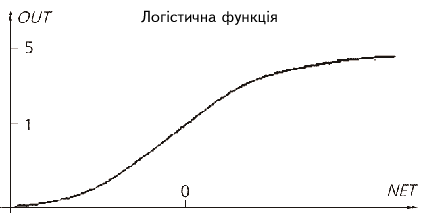


Рис.1.4а. Сигмоїдальна логістична функція

Іншою активаційною функцією, що широко використовується є гіперболічний тангенс. За формою вона схожа з логістичною функцією і часто використовується біологами як математична модель активації нервової клітки. Як активаційна функція штучної нейронної мережі вона записується таким чином:

*OUT = th*(*х*).

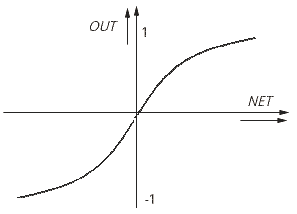


Рис.1.4б. Функція гіперболічного тангенса

Подібно логістичній функції гіперболічний тангенс є *S*-образною функцією, але він симетричний відносно початку координат, і в точці *NET* = 0 значення вихідного сигналу *OUT* дорівнює нулю (див. рис. 1.4б). На відміну від логістичної функції гіперболічний тангенс приймає значення різних знаків, що виявляється вигідним для ряду мереж (див. розділ 3).

Розглянута проста модель штучного нейрона ігнорує багато які властивості свого біологічного двійника. Наприклад, вона не бере до уваги затримки у часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що більш важливо, вона не враховує впливів функції частотної модуляції або синхронізуючої функції біологічного нейрона, які ряд дослідників вважають вирішальними.

Незважаючи на ці обмеження, мережі, побудовані з цих нейронів, виявляють властивості, що сильно нагадують біологічну систему. Тільки час і дослідження зможуть відповісти на питання, чи є подібні збіги випадковими або слідством того, що в моделі вірно схоплені найважливіші межі біологічного нейрона.

**Одношарові штучні нейронні мережі**

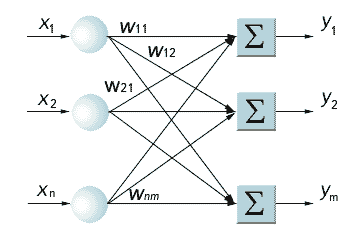


Рис. 1.5. Одношарова нейронна мережа

Хоч один нейрон і здатний виконувати найпростіші процедури розпізнавання, сила нейронних обчислень виникає від з'єднань нейронів в мережах. Найпростіша мережа складається з групи нейронів, що створюють прошарок, як показано в правій частині рис. 1.5. Зазначимо, що вершини-кола зліва служать лише для розподілу вхідних сигналів. Вони не виконують ніяких обчислень, і тому не будуть вважатися прошарком. З цієї причини вони позначені колами, щоб відрізняти їх від обчислювальних нейронів, позначених квадратами. Кожний елемент з множини входів Х окремою вагою сполучений з кожним штучним нейроном. А кожний нейрон видає зважену суму входів в мережу. У штучних і біологічних мережах багато які з'єднання можуть бути відсутнім, всі з'єднання показані з метою спільності. Можуть мати місце також з'єднання між виходами і входами елементів в шарі. Такі конфігурації розглядаються в розділі 6.

Зручно вважати ваги елементами матриці *W*. Матриця має *m* рядків і *n* стовпців, де *m* - число входів, а *n* - число нейронів. Наприклад, *w*23 - це вага, що зв'язує третій вхід з другим нейроном. Таким чином, обчислення вихідного вектора *N*, компонентами якого є виходи *OUT* нейронів, зводиться до матричного множення *N = XW*, де *N* і *Х* - вектори-рядки.

**Багатошарові штучні нейронні мережі**

Більш великі і складні нейронні мережі мають, як правило, і великі обчислювальні можливості. Хоч створені мережі всіх конфігурацій, які тільки можна собі представити, пошарова організація нейронів копіює прошаркові структури певних відділів мозку. Виявилося, що такі багатошарові мережі володіють більшими можливостями, ніж одношарові (див. розділ 2), і в останні роки були розроблені алгоритми для їх навчання.

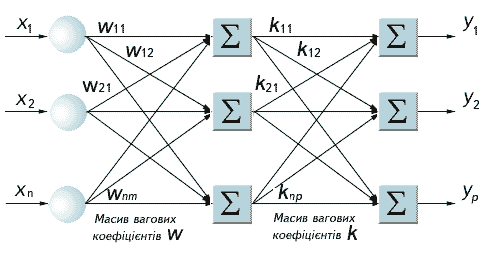


Рис. 1.6. Двошарова нейронна мережа

Багатошарові мережі можуть утворюватися каскадами прошарків. Вихід одного прошарку є входом для подальшого прошарку. Подібна мережа показана на рис. 1.6 і знов зображена з всіма з'єднаннями.

**Нелінійна активаційна функція**

Багатошарові мережі не можуть привести до збільшення обчислювальної потужності в порівнянні з одношаровою мережею лише в тому випадку, якщо активаційна функція між прошарками буде нелінійною. Обчислення виходу прошарку полягає в множенні вхідного вектора на першу вагову матрицю з подальшим множенням (якщо відсутня нелінійна активаційна функція) результуючого вектора на другу вагову матрицю.

(*XW*1)*W*2

Оскільки множення матриць асоціативне, то

*X*(*W*1*W*2).

Це показує, що двошарова лінійна мережа еквівалентна одному прошарку з ваговою матрицею, рівною виробленню двох вагових матриць. Отже, будь-яка багатошарова лінійна мережа може бути замінена еквівалентною одношаровою мережею. У розділі 2 показано, що одношарові мережі досить обмежені по своїх обчислювальних можливостях. Таким чином, для розширення можливостей мереж в порівнянні з одношаровою мережею необхідна нелінійна активаційна функція.

**Мережі із зворотними зв'язками**

У мереж, розглянутих досі, не було зворотних зв'язків, тобто з'єднань, що йдуть від виходів деякого прошарку до входів цього ж прошарку або попередніх шарів. Цей спеціальний клас мереж, званих мережами без зворотних зв'язків або мережами прямого поширення, представляє інтерес і широко використовується. Мережі більш загального вигляду, що мають з'єднання від виходів до входів, називаються мережами із зворотними зв'язками. В мережах без зворотних зв'язків немає пам'яті, їх вихід повністю визначається поточними входами і значеннями ваг. В деяких конфігураціях мереж із зворотними зв'язками попередні значення виходів повертаються на входи; вихід, отже, визначається як поточним входом, так і попередніми виходами. З цієї причини мережам із зворотними зв'язками притаманні властивості, схожі з короткочасною людською пам'яттю, мережеві виходи частково залежать від попередніх входів.

**Термінологія, позначення і схематичне зображення штучних нейронних мереж**

На жаль, для штучних нейронних мереж ще немає опублікованих стандартів і термінів, що устоялися, позначень і графічних представлень. Часом ідентичні мережеві парадигми, представлені різними авторами, здаються далекими між собою. У цій книзі вибрані терміни, що найбільш широко використовуються.

**Термінологія**

Багато які автори уникають терміну "нейрон" для позначення штучного нейрона, вважаючи його досить грубою моделлю свого біологічного прототипу. У цій книзі терміни "нейрон", "клітина", "елемент" використовуються для позначення "штучного нейрона" як короткі і вичерпні.

**Диференційні рівняння або різницеві рівняння**

Алгоритми навчання, як і взагалі штучні нейронні мережі, можуть бути представлені як в диференційній, так і в звичайно-різницевій формі. При використанні диференційних рівнянь вважають, що процеси безперервні і здійснюються подібно великій аналоговій мережі. Для біологічної системи, що розглядається на мікроскопічному рівні, це не так. Активаційний рівень біологічного нейрона визначається середньою швидкістю, з якою він посилає дискретні потенційні імпульси по своєму аксону. Середня швидкість звичайно розглядається як аналогова величина, але важливо не забувати про дійсний стан речей.

Якщо моделювати штучну нейронну мережу на аналоговому комп'ютері, то бажано використати представлення за допомогою диференційних рівнянь. Однак сьогодні більшість робіт виконується на цифрових комп'ютерах, що примушує віддавати перевагу звичайно-різницевій формі як найбільш легко що програмується. З цієї причини протягом всієї книги використовується звичайно-різницеве представлення.

**Графічне представлення**

Як видно з публікацій, немає загальноприйнятого способу підрахунку числа прошарків в мережі. Багатошарова мережа складається, як показано на рис. 1.6, з множин нейронів і ваг. Раніше в зв'язку з рис. 1.5 вже говорилося, що вхідний прошарок не виконує підсумовування. Ці нейрони служать лише як розгалуження для першої множини ваг і не впливають на обчислювальні можливості мережі. З цієї причини перший прошарок не береться до уваги при підрахунку шарів, і мережу, подібна зображеної на рис. 1.6, вважається двошаровою, оскільки тільки два прошарки виконують обчислення. Далі, ваги прошарку вважаються пов'язаними з наступними за ними нейронами. Отже, прошарок складається з множини ваг з наступними за ними нейронами, підсумовуючими зважені сигнали.

**Навчання штучних нейронних мереж**

Серед всіх цікавих властивостей штучних нейронних мереж жодна не захоплює так уяви, як їх здібність до навчання. Їх навчання до такої міри нагадує процес інтелектуального розвитку людської особистості що може показатися, що досягнуте глибоке розуміння цього процесу. Але виявляючи обережність, потрібно стримувати ейфорію. Можливості навчання штучних нейронних мереж обмежені, і треба вирішити багато складних задач для визначення, на чи правильному шляху ми знаходимося. Проте вже отримані переконливі досягнення, такі як "мережа, що розмовляє" Сейновського (див. розділ 3), і створюється багато інших практичних застосувань.

**Мета навчання**

Мережа навчається, щоб для деякої множини входів давати бажану (або, принаймні, подібну на неї) множину виходів. Кожна така вхідна (або вихідна) множина розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних векторів з одночасним налаштуванням ваг відповідно до певної процедури. У процесі навчання ваги мережі поступово стають такими, щоб кожний вхідний вектор виробляв вихідний вектор.

**Навчання з вчителем**

Розрізнюють алгоритми навчання з вчителем і без вчителя. Навчання з вчителем передбачає, що для кожного вхідного вектора існує цільовий вектор, що являє собою необхідний вихід. Разом вони називаються навчальною парою. Переважно мережа навчається на деякому числі таких навчальних пар. Пред'являється вихідний вектор, обчислюється вихід мережі і порівнюється з відповідним цільовим вектором, різниця (похибка) за допомогою зворотного зв'язку подається в мережу і ваги змінюються відповідно до алгоритму, прагнучого мінімізувати похибку. Вектори навчальної множини пред'являються послідовно, обчислюються похибки і ваги налаштовуються для кожного вектора доти, поки похибка по всьому навчальному масиву не досягне прийнятно низького рівня.

**Навчання без вчителя**

Незважаючи на численні прикладні застосування, навчання з вчителем критикувалося за свою біологічну неправдоподібність. Важко уявити навчальний механізм в мозку, який би порівнював бажані і дійсні значення виходів, виконуючи корекцію за допомогою зворотного зв'язку. Якщо допустити подібний механізм в мозку, то звідки тоді виникають бажані виходи? Навчання без вчителя є більш правдоподібною моделлю навчання в біологічній системі. Розвинена Кохоненом [3] і багатьма іншими, вона не потребує цільового вектора для виходів і, отже, не вимагає порівняння з встановленими ідеальними відповідями. Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм налаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних векторів давало однакові виходи. Процес навчання, отже, виділяє статистичні властивості навчальної множини і групує схожі вектори в класи. Пред'явлення на вхід вектора з даного класу дасть певний вихідний вектор, але до навчання неможливо передбачити, який вихід буде вироблятися даним класом вхідних векторів. Отже, виходи подібної мережі повинні трансформуватися в деяку зрозумілу форму, зумовлену процесом навчання. Це не є серйозною проблемою. Переважно, не складно ідентифікувати зв'язок між входом і виходом, встановлений мережею.

**Алгоритми навчання**

Більшість сучасних алгоритмів навчання виросла з концепцій Хеба [2]. Ним запропонована модель навчання без вчителя, в якої синаптична сила (вага) зростає, якщо активовані обидва нейрони, джерело і приймач. Таким чином, що часто використовуються шляхи в мережі підсилюються і феномен звички і навчання через повторення отримує пояснення.

У штучній нейронній мережі, що використовує навчання за Хебом, нарощування ваги визначається виробленням рівнів збудження передаючого і приймаючого нейронів. Це можна записати як

*wij*(*n*+1) = *w*(*n*) + *aOUTi OUTj*,

де *wij*(*n*) - значення ваги від нейрона *i* донейрона *j* до налаштування, *wij*(*n+*1) - значення ваги від нейрона *i* до нейрона *j* після налаштування, a - коефіцієнт швидкості навчання, *OUTi* - вихід нейрона *i* і вхід нейрона *j*, *OUTj* - вихід нейрона *j*.

Мережі, що використовують навчання за Хебом, конструктивно розвивалися, однак за останні 20 років були розвинені більш ефективні алгоритми навчання. Зокрема, в роботах [4 6] і багатьох інших були розвинені алгоритми навчання з вчителем, що приводять до мереж з більш широким діапазоном характеристик навчальних вхідних образів і великими швидкостями навчання, ніж в тих, що використовують просте навчання за Хебом.

У цей час використовується величезна різноманітність навчальних алгоритмів. Була би потрібна значно більша по об'єму книга, чим ця, для розгляду цього предмета повністю. Щоб розглянути цей предмет систематично, якщо і не вичерпне, в кожному з подальших розділів детально описані алгоритми навчання для парадигми, що розглядається в розділі. У доповнення в додатку Б представлений загальний огляд, в певній мірі ширший, хоч і не дуже глибокий. У ньому наданий історичний контекст алгоритмів навчання, їх загальна таксономія, ряд переваг і обмежень. Внаслідок необхідності це приведе до повторення частини матеріалу, виправданням йому служить розширення погляду на предмет.

Оскільки всі штучні нейронні мережі базуються на концепції нейронів, з'єднань та передатних функцій, існує подібність між різними структурами або архітектурами нейронних мереж. Більшість змін походить з різних правил навчання. В цій лекції розглянемо деякі з найвідоміших штучних нейромереж, які організовані в певні категорії застосувань.

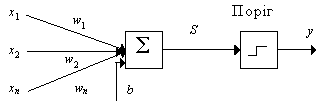
**Перцептрон Розенбалата**

Першою моделлю нейромереж вважають перцептрон Розенбалата. Теорія перцептронів є основою для багатьох типів штучних нейромереж прямого поширення і вони є класикою для вивчення.

Одношаровий перцептрон здатний розпізнавати найпростіші образи. Окремий нейрон обчислює зважену суму елементів вхідного сигналу, віднімає значення зсуву і пропускає результат через жорстку порогову функцію, вихід якої дорівнює +1 чи -1. В залежності від значення вихідного сигналу приймається рішення:

* +1 - вхідний сигнал належить класу A,
* -1 - вхідний сигнал належить класу B.

На рис. 1 показана схема нейронів, використовуваних в одношарових перцептронах, графік передатної функції і схема вирішальних областей, створених у багатовимірному просторі вхідних сигналів. Вирішальні області визначають, які вхідні образи будуть віднесені до класу A, які - до класу B. Перцептрон, що складається з одного нейрона, формує дві вирішальні області, розділені гіперплощиною. На рисунку показаний випадок, коли розмірність вихідного сигналу дорівнює 2. При цьому поділяюча поверхня уявляє собою пряму лінію на площині. Рівняння, що задає поділяючу пряму, залежить від значень синаптичних ваг і зсуву.



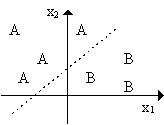
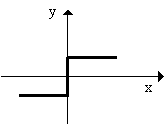


Рис. 1. Схема нейрона, графік передатної функції і поділяюча поверхня

**Алгоритм навчання одношарового перцептрона**

1. Ініціалізація синаптичних ваг і зсуву: синаптичні ваги приймають малі випадкові значення.
2. Пред'явлення мережі нового вхідного і бажаного вихідного сигналів: вхідний сигнал *x*=(*x*1, *x*2,..., *xn*) пред'являється нейрону разом з бажаним вихідним сигналом *d*.
3. Обчислення вихідного сигналу нейрона:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1379.gif

1. Налаштування значень ваг:

*wi*(*t*+1)=*wi* (*t*)+*r*[*d*(*t*)-*y*(*t*)]*xi* (*t*), *i*=1, ..., *N*

http://www.victoria.lviv.ua/html/oio/images/theme7/image1380.gif

де *wі*(*t*) - вага зв'язку від *і*-го елемента вхідного сигналу до нейрона в момент часу *t*, *r* - швидкість навчання (менше 1); *d*(*t*) - бажаний вихідний сигнал.

Якщо мережа приймає правильне рішення, синаптичні ваги не модифікуються.

1. Перехід до кроку 2.

***Тип вхідних сигналів:*** бінарні чи аналогові (дійсні).

***Розмірності входу і виходу*** обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється нейронна мережа, при апаратній реалізації - технологічними можливостями.

***Області застосування:*** розпізнавання образів, класифікація.

***Недоліки.*** Примітивні поділяючі поверхні (гіперплощини) дають можливість вирішувати лише найпростіші задачі розпізнавання.

***Переваги.*** Програмні та апаратні реалізації моделі дуже прості. Простий і швидкий алгоритм навчання.

***Модифікації.*** Багатошарові перцептрони дають можливість будувати більш складні поділяючі поверхні і тому більш поширені.

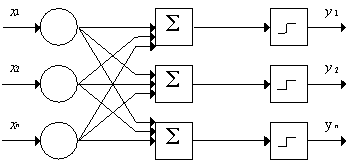


Рис. 2. Перцептрон із багатьма виходами

**Нейромережа зворотного поширення похибки (Back Propagation)**

Архітектура *FeedForward BackPropagation* була розроблена на початку 1970-х років декількома незалежними авторами: Вербор (*Werbor*); Паркер (*Parker*); Румельгарт (*Rumelhart*), Хінтон (*Hinton*) та Вільямс (*Williams*). На даний час, парадигма *ВackРropagation* найбільш популярна, ефективна та легка модель навчання для складних, багатошарових мереж. Вона використовується у різних типах застосувань і породила великий клас нейромереж з різними структурами та методами навчання.

Типова мережа *ВackРropagation* має вхідний прошарок, вихідний прошарок та принаймні один прихований прошарок. Теоретично, обмежень відносно числа прихованих прошарків не існує, але практично застосовують один або два (рис. 3).

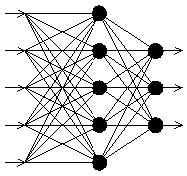


Рис. 3. Багатошаровий перцептрон

Нейрони організовані в пошарову структуру з прямою передачею сигналу. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення. Мережа може моделювати функцію практично будь якої складності, причому число прошарків і число нейронів у кожному прошарку визначають складність функції. Визначення числа проміжних прошарків і числа нейронів в них є важливим при моделюванні мережі. Більшість дослідників та інженерів, застосовуючи архітектуру до визначених проблем використовують загальні правила, зокрема:

1. Кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх прошарків, число нейронів прихованого прошарку *n*прих обирається емпіричним шляхом. В більшості випадків достатня кількість нейронів становить *n*прих Ј *n*вх + *n*вих, де *n*вх, *n*вих - кількість нейронів у вхідному і, відповідно, у вихідному прошарках.
2. Якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого прошарку повинна також збільшитись.
3. Якщо процес, що моделюється, може розділятись на багато етапів, потрібен додатковий прихований прошарок (прошарки). Якщо процес не розділяється на етапи, тоді додаткові прошарки можуть допустити перезапам'ятовування і, відповідно, невірне загальне рішення.

Після того, як визначено число прошарків і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів мережі, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують алгоритми навчання, де відбувається підгонка моделі мережі до наявних навчальних даних. Похибка для конкретної моделі мережі визначається шляхом проходження через мережу всіх навчальних прикладів і порівняння спродукованих вихідних значень з бажаними значеннями. Множина похибок створює функцію похибок, значення якої можна розглядати, як похибку мережі. В якості функції похибок найчастіше використовують суму квадратів похибок.

Для кращого розуміння алгоритму навчання мережі *Back Propagation* потрібно роз'яснити поняття поверхні станів. Кожному значенню синаптичних ваг і порогів мережі (вільних параметрів моделі кількістю *N*) відповідає один вимір в багатовимірному просторі. *N*+1-ий вимір відповідає похибці мережі. Для різноманітних сполучень ваг відповідну похибку мережі можна зобразити точкою в *N*+1-вимірному просторі, всі ці точки утворюють деяку поверхню - поверхню станів. Мета навчання нейромережі полягає в знаходженні на багатовимірній поверхні найнижчої точки.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних мінімумів (точки, найнижчі в своєму певному околі, але вищі від глобального мінімуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального мінімуму на поверхні станів, тому навчання нейромережі по суті полягає в дослідженні цієї поверхні. Відштовхуючись від початкової конфігурації ваг і порогів (від випадково обраної точки на поверхні), алгоритм навчання поступово відшукує глобальний мінімум. Обчислюється вектор градієнту поверхні похибок, який вказує напрямок найкоротшого спуску по поверхні з заданої точки. Якщо трошки просунутись по ньому, похибка зменшиться. Зрештою алгоритм зупиняється в нижній точці, що може виявитись лише локальним мінімумом (в ідеальному випадку - глобальним мінімумом). Складність тут полягає у виборі довжини кроків. При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути рішення, або піти в неправильному напрямку. При маленькому кроці, правильний напрямок буде виявлений, але зростає кількість ітерацій. На практиці розмір кроку береться пропорційним крутизні схилу з деякою константою - швидкістю навчання. Правильний вибір швидкості навчання залежить від конкретної задачі і здійснюється дослідним шляхом. Ця константа може також залежати від часу, зменшуючись по мірі просування алгоритму.

Алгоритм діє ітеративне, його кроки називаються епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні приклади, вихідні значення мережі порівнюються з бажаними значеннями і обчислюється похибка. Значення похибки, а також градієнту поверхні станів використовують для корекції ваг, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня малості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій останову).

**Алгоритм навчання мережі**

1. Ініціалізація мережі: вагові коефіцієнти і зсуви мережі приймають малі випадкові значення.
2. Визначення елемента навчальної множини: (вхід - вихід). Входи (*x*1, *x*2... *xN*), повинні розрізнятися для всіх прикладів навчальної множини.
3. Обчислення вихідного сигналу:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1383.gif

*yim*= *f*(*Sjm*)

*im*=1, 2, ..., *Nm*, *m*=1, 2, ..., *L*

де *S* - вихід суматора, *w* - вага зв'язку, *y* - вихід нейрона, *b* - зсув, *i* - номер нейрона, *N* - число нейронів у прошарку, *m* - номер прошарку, *L* - число прошарків, *f*- передатна функція.

1. Налаштування синаптичних ваг:

*wij*(*t*+1)=*wij*(*t*)+*rgjx'і*

де *wij* - вага від нейрона *i* або від елемента вхідного сигналу *i* до нейрона *j* у момент часу *t*, *xi*' - вихід нейрона *i*, *r* - швидкість навчання, *gj* - значення похибки для нейрона *j*.

Якщо нейрон з номером *j* належить останньому прошарку, тоді

*gj*=*yj*(1-*yj*)(*dj*-*yj*)

де *dj* - бажаний вихід нейрона *j*, *yj* - поточний вихід нейрона *j*.

Якщо нейрон з номером *j* належить одному з прошарків з першого по передостанній, тоді

http://www.victoria.lviv.ua/html/oio/images/theme7/image1384.gif

де *k* пробігає всі нейрони прошарку з номером на одиницю більше, ніж у того, котрому належить нейрон *j*.

Зовнішні зсуви нейронів *b* налаштовуються аналогічним образом.

***Тип вхідних сигналів:*** цілі чи дійсні.

***Тип вихідних сигналів:*** дійсні з інтервалу, заданого передатною функцією нейронів.

***Тип передатної функції:*** сигмоїдальна. В нейронних мережах застосовуються кілька варіантів сигмоїдальних передатних функцій.

*Функція Ферми* (експонентна сигмоїда):

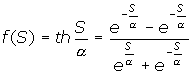
http://www.victoria.lviv.ua/html/oio/images/theme7/image1385.gif

де s - вихід суматора нейрона, a - деякий параметр.

*Раціональна сигмоїда:*

http://www.victoria.lviv.ua/html/oio/images/theme7/image1386.gif

*Гіперболічний тангенс:*



Згадані функції відносяться до однопараметричних. Значення функції залежить від аргументу й одного параметра. Також використовуються багатопараметричні передатні функції, наприклад:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1388.gif

Сигмоїдальні функції є монотонно зростаючими і мають відмінні від нуля похідні по всій області визначення. Ці характеристики забезпечують правильне функціонування і навчання мережі.

***Області застосування.*** Розпізнавання образів, класифікація, прогнозування.

***Недоліки.*** Багатокритеріальна задача оптимізації в методі зворотного поширення розглядається як набір однокритеріальних задач - на кожній ітерації відбуваються зміни значень параметрів мережі, що покращують роботу лише з одним прикладом навчальної вибірки. Такий підхід істотно зменшує швидкість навчання.

***Переваги.*** Зворотне поширення - ефективний та популярний алгоритм навчання багатошарових нейронних мереж, з його допомогою вирішуються численні практичні задачі.

***Модифікації.*** Модифікації алгоритму зворотного поширення зв'язані з використанням різних функцій похибки, різних процедур визначення напрямку і величини кроку.

**Delta Bar Delta**

*Delta bar Delta* була розроблена Робертом Джекобсом (*Robert Jacobs*), для покращення оцінки навчання стандартних мереж *FeedForward* і є модифікацією мережі *BackPropagation*.

Процедура *BackPropagation* базується на підході найкрутішого спуску, який мінімізує похибку передбачення мережі під час процесу змінювання синаптичних ваг. Стандартні оцінки навчання застосовуються на базисі "прошарок за прошарком" і значення моменту призначаються глобально. Моментом вважається фактор, що використовується для згладжування оцінки навчання. Момент додається до стандартної зміни ваги і є пропорційним до попередньої зміни ваги. Хоча цей метод є успішним у вирішенні багатьох задач, збіжність процедури занадто повільна для використання.

Парадигма *Delta bar Delta* має "неформальний" підхід до навчання штучних мереж, при якому кожна вага має свій власний самоадаптований фактор навчання і минулі значення похибки використовуються для обчислення майбутніх значень. Знання ймовірних похибок дозволяє мережі робити інтелектуальні кроки при змінюванні ваг, але процес ускладнюється тим, що кожна вага може мати зовсім різний вплив на загальну похибку. Джекобс запропонував поняття "здорового глузду", коли кожна вага з'єднання мережі повинна мати власну оцінку навчання, а розмір кроку, що призначений до одної ваги з'єднання не застосовується для всіх ваг у прошарку.

Оцінка навчання ваги з'єднання змінюється на основі інформації про біжучу похибку, знайденої із стандартної *ВackРropagation*. Якщо локальна похибка має однаковий знак для декількох послідовних часових кроків, оцінка навчання для цього з'єднання лінійно збільшується. Якщо локальна похибка часто змінює знак, оцінка навчання зменшується геометрично і це гарантує, що оцінки навчання з'єднання будуть завжди додатними. Оцінкам навчання дозволено змінюватись в часі. Призначення оцінки навчання до кожного з'єднання, дозвіл цій оцінці навчання неперервне змінюватись з часом обумовлюють зменшення часу збіжності.

З дозволом різних оцінок навчання для кожної ваги з'єднання у мережі, пошук найкрутішого спуску може не виконуватись. Замість цього, ваги з'єднань змінюються на основі часткових похідних похибки відносно самої ваги і оцінці "кривини поверхні похибки" поблизу біжучої точки значення ваги. Зміни ваг відповідають обмеженню місцевості і вимагають інформацію від нейронів, з якими вони з'єднані.

***Переваги.*** Парадигма *Delta Bar Delta* є спробою прискорити процес збіжності алгоритму зворотного поширення за рахунок використання додаткової інформації про зміну параметрів і ваг під час навчання.

***Недоліки:***

* Навіть невелике лінійне збільшення коефіцієнта може привести до значного росту швидкості навчання, що викликає стрибки в просторі ваг.
* Геометричне зменшення коефіцієнта іноді виявляється не досить швидким.

**Extended Delta Bar Delta**

Елі Мінаї (*Ali Minai*) та Рон Вільямс (*Ron Williams*) розробили алгоритм *Extended Delta bar Delta*, як природній наслідок роботи Джекобса. В алгоритм вбудовується пам'ять з особливістю відновлення. Після кожного представлення навчальних даних епохи, оцінюється накопичена похибка. Якщо похибка є меншою за попередню мінімальну похибку, ваги зберігаються у пам'яті, як найкращі на цей час. Параметр допуску керує фазою відновлення. У випадку, коли біжуча похибка перевищує мінімальну попередню похибку, модифіковану параметром допуску, всі значення ваг з'єднань стохастичне повертаються до збереженої у пам'яті найкращої множини ваг.

**Скерований випадковий пошук**

Мережа скерованого випадкового пошуку (*Directed Random Search*), використовує стандартну архітектуру *FeedForward*, яка не базується на алгоритмі *BackProragation* і коректує ваги випадковим чином. Для забезпечення порядку в такому процесі, до випадкового кроку додається компонента напрямку, яка гарантує скерування ваг до попередньо успішного напрямку пошуку. Вплив на всі нейрони здійснюється окремо.

Для збереження ваг всередині компактної області, де алгоритм працює добре, встановлюють верхню межу величини ваги. Встановлюючи межі ваг великими, мережа може продовжувати працювати, оскільки справжній глобальний оптимум лишається невідомим. Іншою особливістю правила навчання є початкова відмінність у випадковому розподілі ваг. У більшості комерційних пакетів існує рекомендоване розробником число для параметра початкової відмінності.

Парадигма випадкового пошуку має декілька важливих рис. Вона є швидкою та легкою у використанні, найкращі результати отримують, коли початкові ваги знаходяться близько до найкращих ваг. Швидкою парадигма є завдяки тому, що для проміжних нейронів похибки не обчислюються, а обчислюється лише вихідна похибка. Алгоритм надає ефект лише в малій мережі, оскільки при збільшенні числа з'єднань, процес навчання стає довгим та важким.

Існує чотири ключові компоненти мережі з випадковим пошуком. Це є випадковий крок, крок реверсування, скерована компонента та самокорегуюча відмінність.

***Випадковий крок.*** До кожної ваги додається випадкова величина. Вся навчальна множина пропускається через мережу, створюючи "похибку передбачення". Якщо нова загальна похибка навчальної множини є меншою за попередню найкращу похибку передбачення, біжучі значення ваг, які включають випадковий крок стають новою множиною "найкращих" ваг. Біжуча похибка передбачення зберігається як нова, найкраща похибка передбачення.

***Крок реверсування.*** Якщо результати випадкового кроку є гіршими за попередні найкращі, випадкова величина віднімається від початкового значення ваги. Це створює множину ваг, які знаходяться в протилежному напрямку до попереднього випадкового кроку. Якщо загальна "похибка передбачення" є меншою за попередню найкращу похибку, біжучі значення ваг та біжуча похибка передбачення зберігаються як найкращі. Якщо і прямий і зворотній кроки не покращують результат, до найкращих ваг додається повністю нова множина випадкових значень і процес починається спочатку.

***Скерована компонента.*** Для збіжності мережі створюється множина скерованих компонент, отриманих по результатах прямого та зворотного кроків. Скеровані компоненти, що відображають ланку успіхів або невдач попередніх випадкових кроків, додаються до випадкових компонент на кожному кроці процедури і забезпечують елемент "здорового глузду" до пошуку. Доведено, що додавання скерованих компонент забезпечує різке підвищення ефективності алгоритму.

***Самокорегуюча відмінність.*** Визначається параметр початкової відмінності для керування початковим розміром випадкових кроків, що додається до ваг. Адаптивний механізм змінює параметр відмінності, який базується на біжучій оцінці успіху або невдачі. Правило навчання припускає, що біжучий розмір кроків для ваг у правильному напрямку збільшується для випадку декількох послідовних успіхів. Навпаки, якщо воно виявляє декілька послідовних невдач, відмінність зменшується для зменшення розміру кроку.

***Переваги.***Для малих та середніх нейромереж, скерований випадковий пошук надає добрі результати за короткий час. Навчання є автоматичним, вимагає невеликої взаємодії з користувачем.

***Недоліки.***Кількість ваг з'єднань накладає практичні обмеження на розмір задачі. Якщо мережа має більше ніж 200 ваг з'єднань, скерований випадковий пошук може вимагати збільшення часу навчання, але продукувати прийнятні рішення.

**Нейронна мережа вищого порядку або функціонально-пов'язана нейронна мережа**

Функціонально-пов'язані мережі були розроблені Йох-Хан Пао (*Yoh-Han Pao*) і детально описані в його книзі "Адаптивне розпізнавання образів та нейронні мережі" (*Adaptive Pattern Recognition and Neural Networks*). Нейромережа розширює стандартну архітектуру *FeedForward BackPropagation* модифікацією вузлів на вхідному прошарку. Входи комбінуються зрозумілим математичним шляхом за допомогою функцій вищого порядку, таких як квадрати, куби або синуси і розширюють сприйняття мережею заданої проблеми. Із назв цих функцій, вищого порядку або функціонально пов'язаних входів і випливає назва нейромереж.

Існує два основних способи додавання вхідних вузлів. В першому, у модель можуть додаватись перехресні добутки вхідних елементів, це називається вхідним добутком або тензорною моделлю, де кожна компонента вхідного образу перемножується зі всіма компонентами вхідного вектора. Наприклад, для мережі *ВackРropagation* з трьома входами (*A*, *B* і *C*), перехресними добутками будуть *AA, BB, CC, AB, AC* та *BC* (елементи другого порядку). Також можуть додаватись елементи третього порядку, такі як ABC.

Другим методом для додавання вхідних вузлів є функціональне розширення базових входів. У випадку моделі з входами *A, B* і *C*, її можна перетворити у модель нейронної мережі вищого порядку зі входами: *A, B, C, SIN*(*A*), *COS*(*B*), *LOG*(*C*), *MAX*(*A,B,C*) та ін. Повний ефект повинна забезпечити мережа з об'єднанням моделі тензорного та функціонального розширення.

Ніякої нової інформації не додається, але розширене представлення входів робить мережу простішою для навчання. Існують обмеження для цієї моделі. Для перетворення початкових входів необхідна обробка більшої кількості вхідних вузлів, що впливає на швидкодію мережі, тому при розширенні входів має бути враховане поєднання точного рішення та порівняно невеликого часу навчання.

**Мережа Кохонена**

Мережа розроблена Тойво Кохоненом на початку 1980-х рр. і принципово відрізняється від розглянутих вище мереж, оскільки використовує неконтрольоване навчання і навчальна множина складається лише із значень вхідних змінних.

Мережа розпізнає кластери в навчальних даних і розподіляє дані до відповідних кластерів. Якщо в наступному мережа зустрічається з набором даних, несхожим ні з одним із відомих зразків, вона відносить його до нового кластеру. Якщо в даних містяться мітки класів, то мережа спроможна вирішувати задачі класифікації. Мережі Кохонена можна використовувати і в задачах, де класи відомі - перевага буде у спроможності мережі виявляти подібність між різноманітними класами.

Мережа Кохонена має всього два прошарки: вхідний і вихідний, що називають самоорганізованою картою. Елементи карти розташовуються в деякому просторі - як правило двовимірному.

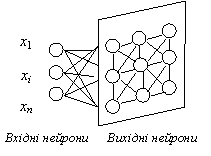


Рис. 4. Мережа Кохонена

Мережа Кохонена навчається методом послідовних наближень. Починаючи з випадковим чином обраного вихідного розташування центрів, алгоритм поступово покращується для кластеризації навчальних даних.

Проте, алгоритм може працювати і на іншому рівні. В результаті ітеративної процедури навчання мережа організовується таким чином, що елементи, які відповідають центрам, розташованим близько один від одного в просторі входів, будуть розташовані близько один від одного і на топологічній карті. Топологічний прошарок мережі можна уявити як двовимірну штахету, яку потрібно так відобразити в *N*-вимірний простір входів, щоб по можливості зберегти вихідну структуру даних. Звісно ж, при будь-якій спробі відтворити *N*-вимірний простір на площині буде загублено багато деталей, але такий прийом дозволяє користувачу візуалізувати дані, що неможливо зрозуміти іншим засобом.

Основний ітераційний алгоритм Кохонена послідовно проходить ряд епох, на кожній епосі опрацьовується один навчальний приклад. Вхідні сигнали - вектори дійсних чисел - послідовно пред'являються мережі. Бажані вихідні сигнали не визначаються. Після пред'явлення достатнього числа вхідних векторів, синаптичні ваги мережі визначають кластери. Крім того, ваги організуються так, що топологічне близькі вузли чуттєві до схожих вхідних сигналів.

Для реалізації алгоритму необхідно визначити міру сусідства нейронів (окіл нейрона-переможця). На мал. 6 показані зони топологічного сусідства нейронів на карті ознак у різні моменти часу. *NEj*(*t*) - множина нейронів, що вважаються сусідами нейрона *j* у момент часу *t*. Зони сусідства зменшуються з часом.

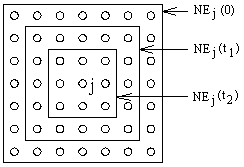


Рис. 5. Зони топологічного сусідства на карті ознак у різні моменти часу

**Алгоритм функціонування мережі Кохонена:**

1. Ініціалізація мережі. Ваговим коефіцієнтам мережі надаються малі випадкові значення. Початкова зона сусідства показана на рис. 5.
2. Пред'явлення мережі нового вхідного сигналу.
3. Обчислення відстані до всіх нейронів мережі:

Відстані *dj* від вхідного сигналу до кожного нейрона *j* визначаються за формулою:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1391.gif

де *xi* - *i-ий* елемент вхідного сигналу в момент часу *t*, *wij*(*t*) - вага зв'язку від *i-го* елемента вхідного сигналу до нейрона *j* у момент часу *t*.

1. Вибір нейрона з найменшою відстанню:

Вибирається нейрон-переможець *j\**, для якого відстань *dj* найменше.

1. Налаштування ваг нейрона *j\** і його сусідів:

Робиться налаштування ваг для нейрона *j\** і всіх нейронів з його околу NE. Нові значення ваг:

*wij*(*t*+1)=*wij*(*t*)+*r*(*t*)(*xi*(*t*)-*wij*(*t*))

де *r*(*t*) - швидкість навчання, що зменшується з часом (додатне число, менше одиниці).

1. Повернення до кроку 2.

В алгоритмі використовується коефіцієнт швидкості навчання, який поступово зменшується, для тонкішої корекції на новій епосі. В результаті позиція центру встановлюється в певній позиції, яка задовільним чином кластеризує приклади, для яких даний нейрон є переможцем.

Властивість топологічної впорядкованості досягається в алгоритмі за допомогою використання поняття околу. Окіл - це декілька нейронів, що оточують нейрон-переможець. Відповідно до швидкості навчання, розмір околу поступово зменшується, так, що спочатку до нього належить досить велике число нейронів (можливо вся карта), на самих останніх етапах окіл стає нульовим і складається лише з нейрона-переможця. В алгоритмі навчання корекція застосовується не тільки до нейрона-переможця, але і до всіх нейронів з його поточного околу. В результаті такої зміни околу, початкові доволі великі ділянки мережі мігрують в бік навчальних прикладів. Мережа формує грубу структуру топологічного порядку, при якій схожі приклади активують групи нейронів, що близько знаходяться на топологічній карті. З кожною новою епохою швидкість навчання і розмір околу зменшуються, тим самим всередині ділянок карти виявляються більш тонкі розходження, що зрештою призводить до точнішого налаштування кожного нейрона. Часто навчання зумисне розбивають на дві фази: більш коротку, з великою швидкістю навчання і великих околів, і більш тривалу з малою швидкістю навчання і нульовими або майже нульовими околами.

Після того, як мережа навчена розпізнаванню структури даних, її можна використовувати як засіб візуалізації при аналізі даних.

***Області застосування.*** Кластерний аналіз, розпізнавання образів, класифікація.

***Недоліки.*** Мережа може бути використана для кластерного аналізу тільки в тому випадку, якщо заздалегідь відоме число кластерів.

***Переваги.*** Мережа Кохонена здатна функціонувати в умовах перешкод, тому що число кластерів фіксоване, ваги модифікуються повільно, налаштування ваг закінчується після навчання.

***Модифікації.*** Одна з модифікацій полягає в тому, що до мережі Кохонена додається мережа MAXNET, що визначає нейрон з найменшою відстанню до вхідного сигналу.

**Квантування навчального вектора (*Learning VectorQuantization*)**

Мережа запропонована Тойво Кохоненом у середині 80-х рр., набагато пізніше за його початкову роботу по самоорганізованим картам. Мережа базується на прошарку Кохонена, який здатний до сортування прикладів у відповідні кластери і використовується як для проблем класифікації, так і для кластеризації зображень.

Мережа містить вхідний прошарок, самоорганізовану карту Кохонена та вихідний прошарок. Приклад мережі зображений на рис. 6. Вихідний прошарок має стільки нейронів, скільки є відмінних категорій або класів. Карта Кохонена має ряд нейронів, згрупованих для кожного з цих класів. Кількість елементів обробки на один клас залежить від складності відношення "вхід-вихід". Звичайно, кожен клас буде мати однакову кількість елементів по всьому прошарку. Прошарок Кохонена навчається класифікації за допомогою навчальної множини. Мережа використовує правила контрольованого навчання. Вхідний прошарок повинен містити лише один нейрон для кожного окремого вхідного параметра.

Квантування навчального вектора класифікує вхідні дані у визначені групування, тобто відображає *n*-вимірний простір у *m*-вимірний простір (бере *n* входів і створює *m* виходів). Карти зберігають відношення між близькими сусідами у навчальній множині так, що вхідні образи, які не були попередньо вивчені, будуть розподілені за категоріями їх найближчих сусідів у навчальних даних.

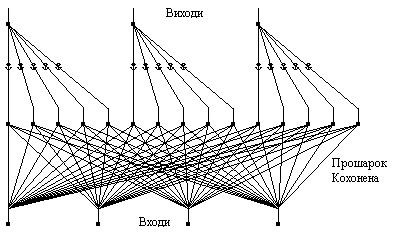


Рис. 6. Приклад мережі з квантуванням навчального вектора.

У режимі навчання, контрольована мережа використовує прошарок Кохонена, де обчислюється відстань від навчального вектора до кожного нейрону і найближчий нейрон оголошується переможцем. Існує лише один переможець на весь прошарок. Переможцю дозволено збуджувати лише один вихідний нейрон, оголошуючи клас або кластер до якого належить вхідний вектор. Якщо нейрон-переможець знаходиться у очікуваному класі навчального вектора, його ваги підсилюються у напрямку навчального вектора. Якщо нейрон-переможець не є у класі навчального вектора, ваги з'єднань зменшуються. Ця остання операція згадується як відштовхування (*repulsion*). Під час навчання окремі нейрони, що приписані до часткового класу мігрують до області, пов'язаної з їх специфічним класом.

Під час режиму функціонування, обчислюється відстань від вхідного вектора до кожного нейрону і знову найближчий нейрон оголошується переможцем. Це в свою чергу генерує один вихід, визначаючи частковий клас, знайдений мережею.

***Недоліки.*** Для складних проблем класифікації з подібними вхідними прикладами, мережа вимагає великої карти Кохонена з багатьма нейронами на клас. Це вибірково може бути подолано вибором доцільних навчальних прикладів або розширенням вхідного прошарку.

Мережа квантування навчального вектора страждає від дефекту, що деякі нейрони мають тенденцію до перемоги занадто часто, тобто налаштовують свої ваги дуже швидко, а інші постійно залишаються незадіяними. Це часто трапляється, коли їх ваги мають значення далекі від навчальних прикладів. Щоб пом'якшити цю проблему, нейрон, який перемагає занадто часто штрафується, тобто зменшуються ваги його зв'язків з кожним вхідним нейроном. Це зменшення ваг є пропорційним до різниці між частотою перемог нейрону та частотою перемог середнього нейрону.

***Переваги.*** Алгоритм граничної корекції використовується для вдосконалення рішення навіть коли було знайдено відносно добре рішення. Алгоритм спроможний діяти, коли нейрон-переможець знаходиться у неправильному класі, а другий найкращий нейрон у правильному класі. Навчальний вектор повинен бути близько від середньої точки простору, що з'єднує ці два нейрони. Неправильний нейрон-переможець зміщується з навчального вектора, а нейрон з іншого місця посувається до навчального вектора. Ця процедура робить чіткішою межу між областями, де можлива невірна класифікація.

На початку навчання бажано відключити відштовхування. Нейрон-переможець пересувається до навчального вектора лише тоді, коли навчальний вектор та нейрон-переможець знаходяться в одному класі. Це право вибору доцільне, коли нейрон повинен оминати область, яка має відмінний клас для досягнення необхідної області.

**Мережа зустрічного поширення (*CounterРropagation*)**

Роберт Хехт-Нільсен (*Robert Hecht-Nielsen*) розробив мережу *СounterРropagation* як засіб для поєднання неконтрольованого прошарку Кохонена із контрольованим вихідним прошарком. Мережа призначена для вирішення складних класифікацій, при мінімізації числа нейронів та часу навчання. Навчання для мережі *СounterРropagation* подібне до мережі з квантуванням навчального вектора.

Приклад мережі зображений на рис. 7. Односкерована мережа *CounterPropagation* має три прошарки: вхідний прошарок, самоорганізовану карту Кохонена та вихідний прошарок, що використовує правило "дельта" для зміни вхідних ваг з'єднань. Цей прошарок називають прошарком Гросберга.

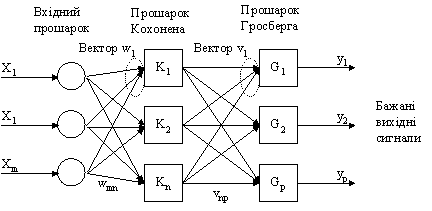


Рис. 7. Мережа зустрічного поширення без зворотних зв'язків

Перша мережа *СounterРropagation* складалась із двоскерованого відображення між вхідним та вихідним прошарками. Дані надходять на вхідний прошарок для генерації класифікації на вихідному прошарку, вихідний прошарок по черзі приймає додатковий вхідний вектор та генерує вихідну класифікацію на вхідному прошарку мережі. Через такий зустрічно-поширений потік інформації випливає назва мережі. Багато розробників використовують односкерований варіант *СounterРropagation*, коли існує лише один шлях прямого поширення від вхідного до вихідного прошарку.

У мережі зустрічного поширення об'єднані два алгоритми: самоорганізована карта Кохонена і зірка Гросберга (*Grossberg Outstar*). Кожен елемент вхідного сигналу подається на всі нейрони прошарку Кохонена. Ваги зв'язків (*wmn*) утворюють матрицю *W*. Кожен нейрон прошарку Кохонена з'єднаний зі всіма нейронами прошарку Гросберга. Ваги зв'язків (*vnp*) утворюють матрицю ваг *V*. Нейронні мережі, що поєднують різні нейропарадигми як будівельні блоки, більш близькі до мозку по архітектурі, ніж однорідні структури. Вважається, що в мозку саме каскадні з'єднання модулів різної спеціалізації дозволяють виконувати необхідні обчислення.

У процесі навчання мережі зустрічного поширення вхідні вектори асоціюються з відповідними вихідними векторами (двійковими або аналоговими). Після навчання мережа формує вихідні сигнали, що відповідають вхідним сигналам. Узагальнююча здатність мережі дає можливість одержувати правильний вихід, якщо вхідний вектор неповний чи спотворений.

**Навчання мережі**

Карта Кохонена класифікує вхідні вектори в групи схожих. В результаті самонавчання прошарок здобуває здатність розділяти несхожі вхідні вектори. Який саме нейрон буде активуватися при пред'явленні конкретного вхідного сигналу, заздалегідь важко передбачити.

При навчанні прошарку Кохонена на вхід подається вхідний вектор і обчислюються його скалярні добутки з векторами ваг всіх нейронів.

Скалярний добуток є мірою подібності між вхідним вектором і вектором ваг. Нейрон з максимальним значенням скалярного добутку з'являється "переможцем" і його ваги підсилюються (ваговий вектор наближається до вхідного).

*wн*=*wc*+*r*(*x*-*wc*)

де *w*н - нове значення ваги, що з'єднує вхідний компонент *x* з нейроном-переможцем, *w*с - попереднє значення цієї ваги, *r* - коефіцієнт швидкості навчання, що спочатку звичайно дорівнює 0.7 і може поступово зменшуватися в процесі навчання. Це дозволяє робити великі початкові кроки для швидкого грубого навчання і менші кроки при підході до остаточної величини.

Кожна вага, зв'язана з нейроном-переможцем Кохонена, змінюється пропорційно різниці між його величиною і величиною входу, до якого він приєднаний. Напрямок зміни мінімізує різницю між вагою і відповідним елементом вхідного прошарку.

Навчальна множина може містити багато подібних між собою вхідних векторів, і мережа повинна бути навченою активувати один нейрон Кохонена для кожного з них. Ваги цього нейрона уявляють собою усереднення вхідних векторів, що його активують.

Виходи прошарку Кохонена подаються на входи нейронів прошарку Гросберга. Входи нейронів обчислюються як зважена сума виходів прошарку Кохонена. Кожна вага коректується лише в тому випадку, якщо вона з'єднана з нейроном Кохонена, який має ненульовий вихід. Величина корекції ваги пропорційна різниці між вагою і необхідним виходом нейрона Гросберга. Навчання прошарку Гросберга - це навчання "з вчителем", алгоритм використовує задані бажані виходи.

**Функціонування мережі**

У своїй найпростішій формі прошарок Кохонена функціонує за правилом "переможець отримує все". Для даного вхідного вектора один і тільки один нейрон Кохонена видає логічну одиницю, всі інші видають нуль.

Прошарок Гросберга функціонує в схожій манері. Його вихід є зваженою сумою виходів прошарку Кохонена.

Якщо прошарок Кохонена функціонує таким чином, що лише один вихід дорівнює одиниці, а інші дорівнюють нулю, то кожен нейрон прошарку Гросберга видає величину ваги, що зв'язує цей нейрон з єдиним нейроном Кохонена, чий вихід відмінний від нуля.

У повній моделі мережі зустрічного поширення є можливість одержувати вихідні сигнали по вхідним і навпаки. Цим двом діям відповідають пряме і зворотне поширення сигналів.

***Області застосування.*** Розпізнавання образів, відновлення образів (асоціативна пам'ять), стиснення даних (із втратами).

***Недоліки.*** Мережа не дає можливості будувати точні апроксимації (точні відображення). У цьому мережа значно уступає мережам зі зворотним поширенням похибки. До недоліків моделі також варто віднести слабкий теоретичний базис модифікацій мережі зустрічного поширення.

***Переваги:***

* Мережа зустрічного поширення проста. Вона дає можливість витягати статистичні властивості з множини вхідних сигналів. Кохонен довів, що для навченої мережі імовірність того, що випадково обраний вхідний вектор буде найближчим до будь-якого заданого вагового вектора, дорівнює *1/k, k* - число нейронів Кохонена.
* Мережа швидко навчається. Час навчання в порівнянні зі зворотним поширенням може бути в 100 разів менше.
* По своїх можливостях будувати відображення мережа зустрічного поширення значно перевершує одношарові перцептрони.
* Мережа корисна для застосувань, у яких потрібно швидка початкова апроксимація.
* Мережа дає можливість будувати функцію і зворотну до неї, що знаходить застосування при рішенні практичних задач.

***Модифікації.*** Мережі зустрічного поширення можуть розрізнятися способами визначення початкових значень синаптичних ваг.

Для підвищення ефективності навчання застосовується додавання шуму до вхідних векторів.

Ще один метод підвищення ефективності навчання - надання кожному нейрону "почуття справедливості". Якщо нейрон стає переможцем частіше, ніж *1/k* (*k* - число нейронів Кохонена), то йому тимчасово збільшують поріг, даючи тим самим навчатися й іншим нейронам.

Крім "методу акредитації", при якому для кожного вхідного вектора активується лише один нейрон Кохонена, може бути використаний "метод інтерполяції", при використанні якого ціла група нейронів Кохонена, що мають найбільші виходи, може передавати свої вихідні сигнали в прошарок Гросберга. Цей метод підвищує точність відображень, реалізованих мережею.

**Імовірнісна нейронна мережа**

Імовірнісна нейронна мережа була розроблена Дональдом Спехтом (*Donald Specht*). Ця мережна архітектура була вперше представлена у двох статтях : "Імовірнісні нейронні мережі для класифікації" (*Probabilistic Neural Networks for Classification*) 1988, "Відображення або асоціативна пам'ять та імовірнісні нейронні мережі" (*Mapping or Associative Memory and Probabilistic Neural Networks*) 1990 р.

Виходи мережі можна інтерпретувати, як оцінки ймовірності належності елементу певному класу. Імовірнісна мережа вчиться оцінювати функцію густини ймовірності, її вихід розглядається як очікуване значення моделі в даній точці простору входів. Це значення пов'язане з густиною ймовірності спільного розподілу вхідних і вихідних даних.

Задача оцінки густини ймовірності відноситься до області байєсівської статистики. Звичайна статистика по заданій моделі показує, яка ймовірність того або іншого виходу (наприклад, на гральній кістці 6 очок буде випадати в середньому в одному випадку з шістьох). Байєсова статистика інтерпретує по іншому: правильність моделі оцінюється по наявних достовірних даних, тобто надає можливість оцінювати густину ймовірності розподілу параметрів моделі по наявних даних.

При рішенні задач класифікації можна оцінити густину ймовірності для кожного класу, порівняти між собою ймовірності приналежності до різних класів і обрати модель з параметрами, при яких густина ймовірності буде найбільшою.

Оцінка густини ймовірності в мережі заснована на ядерних оцінках. Якщо приклад розташований в даній точці простору, тоді в цій точці є певна густина ймовірності. Кластери з близько розташованих точок, свідчать, що в цьому місці густина імовірності велика. Поблизу спостереження є більша довіра до рівня густини, а по мірі віддалення від нього довіра зменшується і плине до нуля. В методі ядерних оцінок в точку, що відповідає кожному прикладу, поміщається деяка проста функція, потім вони всі додаються і в результаті утворюється оцінка для загальної густини імовірності. Найчастіше в якості ядерних функцій беруть дзвоноподібні функції (гаусові). Якщо є достатня кількість навчальних прикладів, такий метод дає добрі наближення до істинної густини імовірності.

Імовірнісна мережа має три прошарки: вхідний, радіальний та вихідний. Радіальні елементи беруться по одному на кожний приклад. Кожний з них містить гаусову функцію з центром в цьому прикладі. Кожному класу відповідає один вихідний елемент. Вихідний елемент з'єднаний лише з радіальними елементами, що відносяться до його класу і підсумовує виходи всіх елементів, що належать до його класу. Значення вихідних сигналів утворюються пропорційно ядерних оцінок ймовірності приналежності відповідним класам.

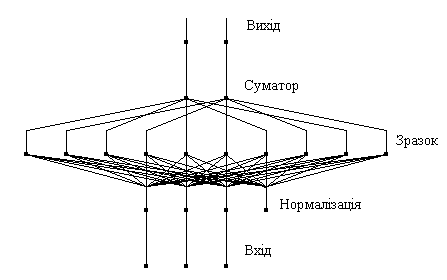


Рис. 8. Приклад імовірнісної нейронної мережі

Базова модель імовірнісної мережі має модифікації. Припустимо, що пропорції класів у навчальній множині відповідають їх пропорціям у всій досліджуваній вибірці (апріорна імовірність). Наприклад, якщо серед всіх людей хворими є 2%, то в навчальній множині для мережі, яка діагностує захворювання, хворих також повинно бути 2%. Якщо ж апріорні імовірності відрізняються від пропорції в навчальній виборці, мережа буде видавати невірний результат. Це можна врахувати, вводячи корегуючі коефіцієнти для різноманітних класів.

Навчання імовірнісної нейронної мережі є набагато простішим, ніж ВackРropagation. Істотним недоліком мережі є її розмір, оскільки вона фактично вміщує в собі всі навчальні дані, потребує багато пам'яті і може повільно працювати.

**Мережа Хопфілда**

Джон Хопфілд вперше представив свою асоціативну мережу у 1982 р. у Національній Академії Наук. На честь Хопфілда та нового підходу до моделювання, ця мережна парадигма згадується як мережа Хопфілда. Мережа базується на аналогії фізики динамічних систем. Початкові застосування для цього виду мережі включали асоціативну, або адресовану за змістом пам'ять та вирішували задачі оптимізації.

Мережа Хопфілда використовує три прошарки: вхідний, прошарок Хопфілда та вихідний прошарок. Кожен прошарок має однакову кількість нейронів. Входи прошарку Хопфілда під'єднані до виходів відповідних нейронів вхідного прошарку через змінні ваги з'єднань. Виходи прошарку Хопфілда під'єднуються до входів всіх нейронів прошарку Хопфілда, за винятком самого себе, а також до відповідних елементів у вихідному прошарку. В режимі функціонування, мережа скеровує дані з вхідного прошарку через фіксовані ваги з'єднань до прошарку Хопфілда. Прошарок Хопфілда коливається, поки не буде завершена певна кількість циклів, і біжучий стан прошарку передається на вихідний прошарок. Цей стан відповідає образу, вже запрограмованому у мережу.

Навчання мережі Хопфілда вимагає, щоб навчальний образ був представлений на вхідному та вихідному прошарках одночасно. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань. Недвійкова реалізація мережі повинна мати пороговий механізм у передатній функції. Для правильного навчання мережі відповідні пари "вхід-вихід" мають відрізнятися між собою.

Якщо мережа Хопфілда використовується як пам'ять, що адресується за змістом вона має два головних обмеження. По-перше, число образів, що можуть бути збережені та точно відтворені є строго обмеженим. Якщо зберігається занадто багато параметрів, мережа може збігатись до нового неіснуючого образу, відмінному від всіх запрограмованих образів, або не збігатись взагалі. Межа ємності пам'яті для мережі приблизно 15% від числа нейронів у прошарку Хопфілда. Другим обмеженням парадигми є те, що прошарок Хопфілда може стати нестабільним, якщо навчальні приклади є занадто подібними. Зразок образу вважається нестабільним, якщо він застосовується за нульовий час і мережа збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором навчальних прикладів більш ортогональних між собою.

Структурна схема мережі Хопфилда приведена на рис. 9.

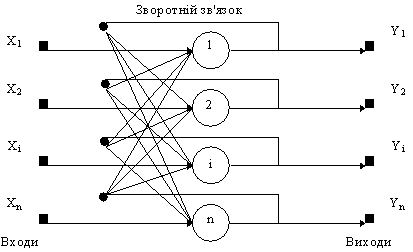


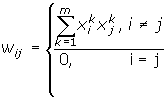
Рис. 9. Структурна схема мережі Хопфілда.

Задача, розв'язувана даною мережею в якості асоціативної пам'яті, як правило, формулюється таким чином. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують якийсь об'єкти або характеристики процесів), вважають зразковим. Мережа повинна вміти з зашумленого сигналу, поданого на її вхід, виділити ("пригадати" по частковій інформації) відповідний зразок або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків. У загальному випадку, будь-який сигнал може бути описаний вектором *x*1, *хі*, *хn*..., *n* - число нейронів у мережі і величина вхідних і вихідних векторів. Кожний елемент *xi* дорівнює або +1, або -1. Позначимо вектор, що описує *k*-ий зразок, через *Xk*, а його компоненти, відповідно, - *xik*, *k*=0, ..., *m*-1, *m* - число зразків. Якщо мережа розпізнає (або "пригадує") якийсь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто *Y = Xk*, де *Y* - вектор вихідних значень мережі: *y*1*, yi, yn*. У противному випадку, вихідний вектор не співпаде з жодний зразковим.

Якщо, наприклад, сигнали являють собою якесь зображення, то, відобразивши у графічному виді дані з виходу мережі, можна буде побачити картинку, що цілком збігається з однієї зі зразкових (у випадку успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

**Алгоритм функціонування мережі**

1. На стадії ініціалізації мережі вагові коефіцієнти синапсів встановлюються таким чином:



Тут *i* і *j* - індекси, відповідно, предсинаптичного і постсинаптичного нейронів; *xik*, *xjk* - *i*-ий і *j*-ий елементи вектора *k*-ого зразка.

1. На входи мережі подається невідомий сигнал (t - номер ітерації). Його поширення безпосередньо встановлює значення виходів:

*yi(0) = xi , i = 0...n-1,*

тому позначення на схемі мережі вхідних сигналів у явному виді носить чисто умовний характер. Нуль у скобці справа від *yi* означає нульову ітерацію в циклі роботи мережі.

1. Розраховується новий стан нейронів

http://www.victoria.lviv.ua/html/oio/images/theme7/image1397.gif, j=0...n-1

і нові значення виходів

http://www.victoria.lviv.ua/html/oio/images/theme7/image1398.gif

де *f* - передатна функція у виді порогової, приведена на рис. 10.

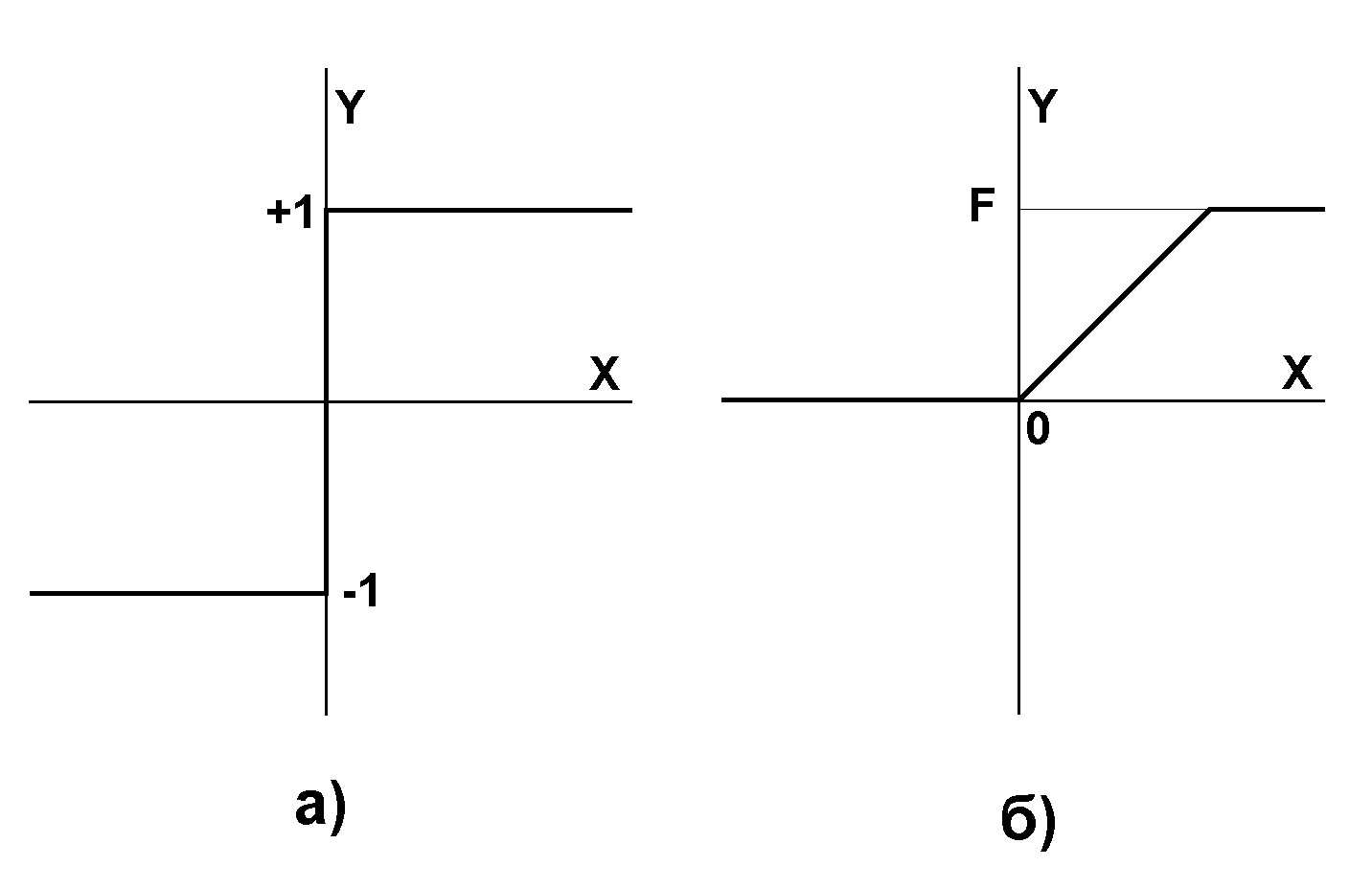


Рис. 10. Передатні функції

1. Перевіряємо чи змінилися вихідні значення виходів за останню ітерацію. Якщо так - перехід до пункту 2, інакше (якщо виходи стабілізувались) - кінець. При цьому вихідний вектор являє собою зразок, що найкраще відповідає вхідним даним.

Іноді мережа не може провести розпізнавання і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей мережі. Для мережі Хопфилда число запам'ятованих образів *m* не повинно перевищувати величини, приблизно рівної 0.15•*n*. Крім того, якщо два образи А и Б сильно схожі, вони, можливо, будуть викликати в мережі перехресні асоціації, тобто пред'явлення на входи мережі вектора А призведе до появи на її виходах вектори Б и навпаки.

**Машина Больцмана**

Машина Больцмана (*Boltzmann mashine*) є подібною за функцією та дією до мережі Хопфілда і включає поняття "модельованого відпалу" для пошуку в просторі станів прошарку образів глобальний мінімум.

Еклі (*Ackley*), Хінтон (*Hinton*) та Сейновскі (*Sejnowski*) розробили правило больцманівського навчання у 1985 р. Подібно до мережі Хопфілда, машина Больцмана має простір станів, який базується на вагах з'єднань у прошарку образів. Процеси навчання мережі, наповненої образами, включає вивчення рельєфу простору станів. Під час ітеративного навчання знаходяться краща множина вихідних значень.

В процесі навчання машина Больцмана моделює відпал металу. Як і при фізичному відпалі, температура починається з вищих значень і зменшується з часом. Збільшена температура додає збільшений шумовий коефіцієнт до кожного нейрону у прошарку образів. Звичайно, кінцева температура є нулем. Для досягнення кращого рішення доцільно на нижчих температурах додавати більше ітерацій.

Машина Больцмана, навчаючись на високій температурі, веде себе більш подібно до випадкової моделі, а на низьких температурах вона веде себе як детермінована модель. Через випадкову компоненту у відпаловому навчанні, нейрон може прийняти нове значення стану, що збільшується швидше, ніж зменшується загальний простір станів. Імітація фізичного відпалу дозволяє уникаючи локальний мінімум, просуватись до глобального.

Як і в мережі Хопфілда, мережі може бути представлений частковий образ для доповнення відсутньої інформації. Обмеження на число класів, що є менше ніж 15 % від загальної кількості елементів у прошарку образів, все ще застосовується.

**Алгоритм функціонування мережі**

1. Визначити змінну *T*, що представляє штучну температуру.
2. Пред'явити мережі множину входів і обчислити виходи та цільову функцію.
3. Дати випадкову зміну вагам і перерахувати вихід мережі та зміну цільової функції у відповідності зі зробленою зміною ваг.
4. Якщо цільова функція зменшилась, то зберегти зміну ваг.

Якщо зміна ваг приводить до збільшення цільової функції, то імовірність збереження цієї зміни обчислюється за допомогою розподілу Больцмана:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1400.gif

де *P*(*c*) - імовірність зміни *c* у цільовій функції; *k* - константа, аналогічна константі Больцмана, що вибирається в залежності від задачі; *T* - штучна температура.

Вибирається випадкове число *r* з рівномірного розподілу від нуля до одиниці. Якщо *P*(*c*) більше, ніж *r*, то зміна зберігається, у противному випадку величина ваги повертається до попереднього значення.

Ця процедура дає можливість системі робити випадковий крок у напрямку, що псує цільову функцію, дозволяючи їй тим самим вириватися з локальних мінімумів.

Кроки 3 і 4 повторюються для кожної з ваг мережі, поступово зменшуючи температуру *T*, поки не буде досягнуте припустиме низьке значення цільової функції. У цей момент пред'являється інший вхідний вектор і процес навчання повторюється. Мережа навчається на всіх векторах навчальної множини, поки цільова функція не стане припустимої для всіх з них.

Швидкість зменшення температури повинна бути зворотно пропорційна логарифму часу. При цьому мережа збігається до глобального мінімуму.

***Області застосування.*** Розпізнавання образів, класифікація.

***Недоліки.*** Повільний алгоритм навчання.

***Переваги.***Алгоритм дає можливість мережі вибиратися з локальних мінімумів адаптивного рельєфу простору станів.

***Модифікації.*** Випадкові зміни можуть проводитися не тільки для окремих ваг, але і для всіх нейронів прошарків в багатошарових мережах або для всіх нейронів мережі одночасно. Ці модифікації алгоритму дають можливість скоротити загальне число ітерацій навчання.

**Мережа Хемінга**

Мережа Хемінга (*Hamming*) є розширенням мережі Хопфілда. Ця мережа була розроблена Річардом Ліппманом (*Richard Lippman*) у середині 80-х рр. Мережа Хемінга реалізує класифікатор, що базується на найменшій похибці для векторів двійкових входів, де похибка визначається відстанню Хемінга. Відстань Хемінга визначається як число бітів, які відрізняються між двома відповідними вхідними векторами фіксованої довжини. Один вхідний вектор є незашумленим прикладом образу, інший є спотвореним образом. Вектор виходів навчальної множини є вектором класів, до яких належать образи. У режимі навчання вхідні вектори розподіляються до категорій для яких відстань між зразковими вхідними векторами та біжучим вхідним вектором є мінімальною.

Мережа Хемінга має три прошарки: вхідний прошарок з кількістю вузлів, скільки є окремих двійкових ознак; прошарок категорій (прошарок Хопфілда), з кількістю вузлів, скільки є категорій або класів; вихідний прошарок, який відповідає числу вузлів у прошарку категорій.

Мережа є простою архітектурою прямого поширення з вхідним рівнем, повністю під'єднаним до прошарку категорій. Кожен елемент обробки у прошарку категорій є зворотно під'єднаним до кожного нейрона у тому ж самому прошарку і прямо під'єднаним до вихідного нейрону. Вихід з прошарку категорій до вихідного прошарку формується через конкуренцію.

Навчання мережі Хемінга є подібним до методології Хопфілда. На вхідний прошарок надходить бажаний навчальний образ, а на виході вихідного прошарку надходить значення бажаного класу, до якого належить вектор. Вихід містить лише значення класу до якої належить вхідний вектор. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань.

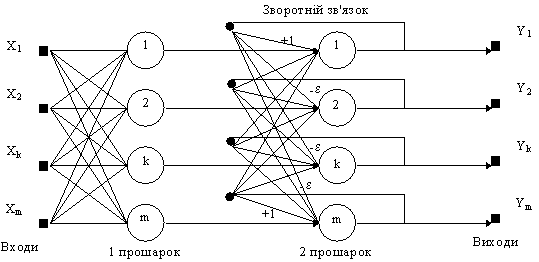


Рис. 11. Структурна схема мережі Хемінга

**Алгоритм функціонування мережі Хемінга**

1. На стадії ініціалізації ваговим коефіцієнтам першого прошарку і порогу передатної функції присвоюються такі значення:

*Wik*=*xIk*/2, i=0...n-1, k=0...m-1

*bk* = *n / 2, k = 0...m-1*

Тут *xik* - *i*-ий елемент *k*-ого зразка.

Вагові коефіцієнти гальмуючих синапсів у другому прошарку беруть рівними деякій величині 0 < *v* < 1/m. Синапс нейрона, пов'язаний із його ж виходом має вагу +1.

1. На входи мережі подається невідомий вектор *x*1*, xi*, *xn* *...*Розраховуються стани нейронів першого прошарку (верхній індекс у скобках указує номер прошарку):

http://www.victoria.lviv.ua/html/oio/images/theme7/image1403.gif*, j=0...m-1*

Після цього отримання значення ініціалізують значення виходів другого прошарку:

*yj(2) = yj(1), j = 0...m-1*

1. Обчислюються нові стани нейронів другого прошарку:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1404.gif

і значення їх виходів:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1405.gif

Передатна функція *f* має вид порога, причому величина *b* повинна бути достатньо великою, щоб будь-які можливі значення аргументу не призводили до насичення.

1. Перевіряється, чи змінилися виходи нейронів другого прошарку за останню ітерацію. Якщо так - перейти до кроку 3. Інакше - кінець.

Роль першого прошарку є умовною: скориставшись один раз на першому кроці 1 значеннями його вагових коефіцієнтів, мережа більше не вертається до нього, тому перший прошарок може бути взагалі виключений із мережі.

Мережа Хемінга має ряд переваг над мережею Хопфілда. Вона реалізує оптимальний класифікатор мінімуму похибки, якщо похибки вхідних бітів є випадковими та незалежними. Для функціонування мережі Хемінга потрібна менша кількість нейронів, оскільки середній прошарок вимагає лише один нейрон на клас, замість нейрону на кожен вхідний вузол. І, нарешті, мережа Хемінга не страждає від неправильних класифікацій, які можуть трапитись у мережі Хопфілда. В цілому, мережа Хемінга є як швидшою, так і точнішою за мережу Хопфілда.

**Двоскерована асоціативна пам'ять**

Ця мережна модель була розроблена Бартом Козко (*Bart Kosko*) і розширює модель Хопфілда. Множина парних образів навчається за образами, що представлені як біполярні вектори. Подібно до мережі Хопфілда, коли представляється зашумлена версія одного образу, визначається найближчий образ, асоційований з ним.

На рис. 12 показаний приклад двоскерованої асоціативної пам'яті. Вона має стільки входів, скільки є вихідних нейронів. Два приховані прошарки містяться на двох окремих асоціативних елементах пам'яті і представляють подвоєний розмір вхідних векторів. Середні прошарки повністю з'єднуються один з одним. Вхідний та вихідний прошарки потрібні для реалізації засобів введення та відновлення інформації з мережі.

Середні прошарки розроблені для збереження асоційованих пар векторів. Коли зашумлений вектор образу вважається вхідним, середні прошарки коливаються до досягнення стабільного стану рівноваги, який відповідає найближчій навченій асоціації і буде генерувати початковий навчальний образ на виході. Подібно до мережі Хопфілда, двоскерована асоціативна пам'ять є схильною до неправильного відшукування навченого образу, якщо надходить невідомий вхідний вектор, який не був у складі навчальної множини.

Двоскерована асоціативна пам'ять відноситься до гетероасоціативної пам'яті. Вхідний вектор надходить на один набір нейронів, а відповідний вихідний вектор продукується на іншому наборі нейронів. Вхідні образи асоціюються з вихідними.

Для порівняння: мережа Хопфилда є автоасоціативною. Вхідний образ може бути відновлений чи виправлений мережею, але не може бути асоційований з іншим образом. У мережі Хопфілда використовується одношарова структура асоціативної пам'яті, у якій вихідний вектор з'являється на виході тих же нейронів, на які надходить вхідний вектор.

Двоскерована асоціативна пам'ять, як і мережа Хопфілда, здатна до узагальнення, виробляючи правильні вихідні сигнали, незважаючи на спотворені входи.

Розглянемо схему двоскерованої асоціативної пам'яті. Вхідний вектор *A* обробляється матрицею ваг *W* мережі, у результаті чого продукується вектор вихідних сигналів мережі *B*. Вектор *B* обробляється транспонованою матрицею *W*T ваг мережі, яка продукує сигнали, що представляють новий вхідний вектор *A*. Цей процес повторюється доти, поки мережа не досягне стабільного стану, у якому ні вектор *A*, ні вектор *B* не змінюються.

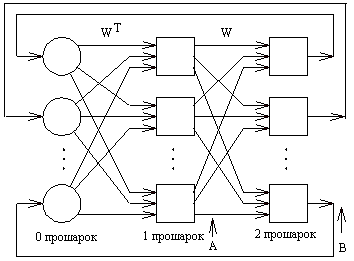


Рис. 12. Двоскерована асоціативна пам'ять

Нейрони в прошарках 1 і 2 функціонують, як і в інших парадигмах, обчислюючи суму зважених входів і значення передатної функції *F*:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1407.gif

або у векторній формі:

*B*=*F*(*AW*)

де *B* - вектор вихідних сигналів нейронів прошарку 2, *A* - вектор вихідних сигналів нейронів прошарку 1, *W* - матриця ваг зв'язків між прошарками 1 і 2, *F* - передатна функція.

Аналогічно *A*=*F*(*BWT*), де *WT* є транспозицією матриці *W*.

В якості передатної функції використовується експонентна сигмоїда.

Прошарок 0 не робить обчислень і не має пам'яті. Він є лише засобом розподілу вихідних сигналів прошарку 2 до елементів матриці *WT*.

Формула для обчислення значень синаптичних ваг:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1408.gif

де *Aj* і *Bj* - вхідні і вихідні сигнали навчальної вибірки.

Вагова матриця обчислюється як сума добутків всіх векторних пар навчальної вибірки.

Системи зі зворотним зв'язком мають тенденцію до коливань. Вони можуть переходити від стану до стану, ніколи не досягаючи стабільності. Доведено, що двоскерована асоціативна пам'ять безумовно стабільна при будь-яких значеннях ваг мережі.

***Області застосування.*** Асоціативна пам'ять, розпізнавання образів.

***Недоліки.*** Ємність двоскерованої асоціативної пам'яті жорстко обмежена. Якщо *n* - кількість нейронів у вхідному прошарку, то число векторів, що можуть бути запам'ятовані в мережі не перевищує *L*=*n*/2*log*2*n*. Так, якщо *n*=1024, то мережа здатна запам'ятати не більш 25 образів.

Двоскерована асоціативна пам'ять має деяку непередбачуваність у процесі функціонування, можливі помилкові відповіді.

***Переваги:***

* У порівнянні з автоасоціативною пам'яттю (наприклад, мережею Хопфилда), двоскерована асоціативна пам'ять дає можливість будувати асоціації між векторами *A* і *B*, що у загальному випадку мають різні розмірності. За рахунок таких можливостей гетероасоціативна пам'ять має більш широкий клас застосувань, ніж автоасоціативна пам'ять.
* Двоскерована асоціативна пам'ять - проста мережа, що може бути реалізована у виді окремої СБИС чи оптоелектронним способом.
* Процес формування синаптичних ваг простий і швидкий. Мережа швидко збігається в процесі функціонування.

***Модифікації:***

* Сигнали в мережі можуть бути як дискретними, так і аналоговими. Для обох випадків доведена стабільність мережі.
* Були запропоновані моделі двоскерованої асоціативної пам'яті з навчанням без вчителя (адаптивна двоскерована асоціативна пам'ять).
* Введення латеральних зв'язків всередині прошарку дає можливість реалізувати конкуруючу двоскеровану асоціативну пам'ять.

**Мережа адаптивної резонансної теорії**

Розроблена Стівеном Гросбергом та Карпентером у середині 80-х рр. Парадигма використовує функцію неконтрольованого навчання і аналізує значні вхідні дані, виявляє можливі ознаки та класифікує образи у вхідному векторі.

Мережа адаптивної резонансної теорії складається з двох взаємопов'язаних прошарків нейронів, розташованих між вхідним та вихідним прошарками. Кожен вхідний образ нижчого прошарку резонансу стимулювати очікуваний образ на вищому прошарку, який пересилається знов до нижчого прошарку, щоб впливати на наступний вхід. Це створює "резонанс" між нижчим та вищим прошарками для полегшення мережної адаптації образів.

Мережа переважно використовується у біологічному моделюванні, проте існують деякі технічні застосування. Головним обмеженням мережної архітектури є її шумова чутливість. Навіть мала кількість шуму на вхідному векторі плутає узагальнюючі можливості навченої мережі.

Мережа ART-1 навчається без учителя і реалізує алгоритм кластеризації, дуже схожий на алгоритм "послідовного лідера". Відповідно до цього алгоритму перший вхідний сигнал вважається зразком першого кластера. Наступний вхідний сигнал порівнюється зі зразком першого кластера. Говорять, що вхідний сигнал "прямує за лідером" і належить першому кластеру, якщо відстань до зразка першого кластера менше порога. У противному випадку другий вхідний сигнал - зразок другого кластера. Цей процес повторюється для всіх наступних вхідних сигналів. Таким чином, число кластерів росте з часом і залежить як від значення порога, так і від метрики відстані, що використовується для порівняння вхідних сигналів і зразків класів.

Основна частина мережі ART-1 схожа з мережею Хемінга. За допомогою послідовних зв'язків обчислюється відповідність вхідних сигналів і зразків кластерів. Максимальне значення відповідності підсилюється за допомогою латеральних зв'язків вихідних нейронів.

Мережа ART-1 відрізняється від мережі Хемінга зворотними зв'язками від вихідних нейронів до вхідних, крім того є можливість виключати вихідний нейрон з максимальним значенням відповідності і проводити тестування відповідності вхідного сигналу і зразків кластерів, як того вимагає алгоритм "послідовного лідера".

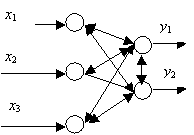


Рис. 13. Основні компоненти класифікатора Карпентер/Гросберга

**Алгоритм функціонування мережі**

1. Ініціалізація мережі:

*vij*(0)=1;*Wij*(0)=1/(1+*N*)

0Ј *i* Ј *N*-1; 0Ј *j* Ј *M*-1; 0Ј *b* Ј 1;

де *wij*(*t*) - синаптична вага зв'язку від *i-го* нейрона першого прошарку до *j-го* нейрона другого прошарку в момент часу *t*, *vij*(*t*) - синаптична вага зв'язку від *i-го* нейрона другого прошарку до *j-го* нейрона першого прошарку в момент часу *t*, *b* - значення порога.

Ваги *vij*(*t*) і *wij*(*t*) визначають зразок, що відповідає нейрону *j*.

Поріг *b* показує, наскільки повинен вхідний сигнал збігатися з одним із запам'ятованих зразків, щоб вони вважалися схожими. Близьке до одиниці значення порога вимагає майже повного збігу.

При малих значеннях порога вхідний сигнал і зразок, навіть якщо вони сильно розрізняються вважаються приналежними одному кластеру.

1. Пред'явлення мережі нового бінарного вхідного сигналу:

Вхідні сигнали пред'являються вихідному прошарку нейронів аналогічно тому, як це робиться в мережі Хемінга.

1. Обчислення значень відповідності:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1412.gif

Значення відповідності обчислюються паралельно для всіх зразків, запам'ятованих у мережі, аналогічно мережі Хемінга.

1. Вибір зразка з найбільшою відповідністю:

*yj*=*max*(*yj*)

Ця операція виконується за допомогою латерального гальмування.

1. Порівняння з порогом:

http://www.victoria.lviv.ua/html/oio/images/theme7/image1413.gif; http://www.victoria.lviv.ua/html/oio/images/theme7/image1414.gif

якщо http://www.victoria.lviv.ua/html/oio/images/theme7/image1415.gifперехід до кроку 7, інакше до кроку 6.

На цьому кроці обчислюється відношення скалярного добутку вхідного сигналу і зразка з найбільшим значенням відповідності до числа одиничних біт вхідного сигналу. Значення відношення порівнюється з порогом, введеному на першому кроці.

Якщо значення відношення більше порога, то вхідний сигнал вважається схожим на зразок з найбільшим значенням відповідності. У цьому випадку зразок модифікується шляхом виконання операції AND (логічне "I"). Новий зразок є зразок на попередньому кроці + вхідний сигнал.

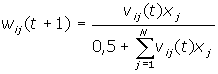
Якщо значення відношення менше порога, то вважається, що вхідний сигнал відрізняється від всіх зразків і він розглядається як новий зразок. У мережу вводиться нейрон, що відповідає новому зразку, і обчислюються значення синаптичних ваг.

1. Виключення приклада з найбільшим значенням відповідності:

Вихід нейрона з найбільшим значенням відповідності тимчасово встановлюється рівним нулю і більш не приймає участь у кроці 4.

1. Адаптація приклада з найбільшим значенням відповідності:

*vij*(*t*+1)=*vij*(*t*)*xj*



1. Включення всіх виключених на кроці 6 зразків. Повернення до кроку 2.

***Вхідні сигнали*** в цій моделі бінарні.

***Розмірності входу і виходу*** обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється нейронна мережа, при апаратній реалізації - технологічними можливостями.

***Ємність мережі*** збігається з числом нейронів другого прошарку і може збільшуватися в процесі функціонування мережі.

***Області застосування.*** Розпізнавання образів, кластерний аналіз.

***Недоліки****.* Необмежене збільшення числа нейронів у процесі функціонування мережі. У присутності шуму виникають значні проблеми, зв'язані з неконтрольованим ростом числа зразків.

***Переваги****.* Навчання без учителя.

***Модифікації.*** Існує модель ART-2 з аналоговими значеннями вхідними сигналами.

**Висновки**

У подальших розділах представлені і проаналізовані деякі найбільш важливі мережеві конфігурації і їх алгоритми навчання. Представлені парадигми дають уявлення про мистецтво конструювання мереж в цілому, його минулому і теперішньому часі. Багато інших парадигм при ретельному розгляді виявляються лише їх модифікаціями. Сьогоднішній розвиток нейронних мереж швидше еволюційний, ніж революційний. Тому розуміння представлених в даній книзі парадигм дозволить стежити за прогресом в цій області, що швидко розвивається.

Натяк зроблений на інтуїтивні і алгоритмічні, а не математичні аспекти. Книга адресована швидше користувачеві штучних нейронних мереж, ніж теоретику. Також представлено досить інформації, щоб дати читачеві можливість розуміти основні ідеї. Ті, хто знайомий з програмуванням, зможуть реалізувати будь-яку з цих мереж. Складні математичні викладення опущені, якщо тільки вони не мають прямого відношення до реалізації мережі. Для зацікавленого читача приводяться посилання на більш точні і повні роботи.

**Література**

1. Grossberg S. 1973. Contour enhancement, short-term memory, and consistencies in reverberating neural networks. Studies in Applied Mathematics 52:217,257.
2. Hebb D. 0. 1961. Organization of behavior. New York: Science Edition.
3. Kohonen Т. 1984. Self-organization and associative memory. Series in Information Sciences, vol. 8. Berlin: Springer Verlag.
4. Rosenblatt F. 1962. Principles of neurodynamics. New York: Spartan Books. (Російський переклад: Розенблатт Ф. Принципы нейродинамики. М.: Мир., 1965.)
5. Widrow В. 1959. Adaptive sampled-data systems, а statistical theory of adaptation. 1959 IRE WESCON Convention Record, part 4, pp. 88-91. New York: Institute of Radio Engineers.
6. Widrow В., Hoff М. 1960. Adaptive switching circuits. I960 IRE WESCON Convention Record, pp. 96-104. New York: Institute of Radio Engineers.