Лекція 7. ВИРІШУВАЧІ ПРОБЛЕМ, ЗАСНОВАНІ НА ЗНАННЯХ. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

1. Основні поняття
2. Типи ШНМ.
3. Навчання ШНМ

1. Основні поняття та типи штучних нейронних мереж

У історії досліджень в області нейронних мереж (НМ), як і в історії будь-якої іншої науки, були свої успіхи і невдачі. Крім того, тут постійно позначається психологічний чинник, що виявляється в нездатності людини описати словами те, як вона думає.

Здібність нейронної мережі до навчання вперше досліджена Дж. Маккалоком і У. Піттом. У 1943 році вийшла їх робота “Логічне числення ідей, що відносяться до нервової діяльності”, в якій була побудована модель нейрона, і сформульовані принципи побудови штучних нейронних мереж.

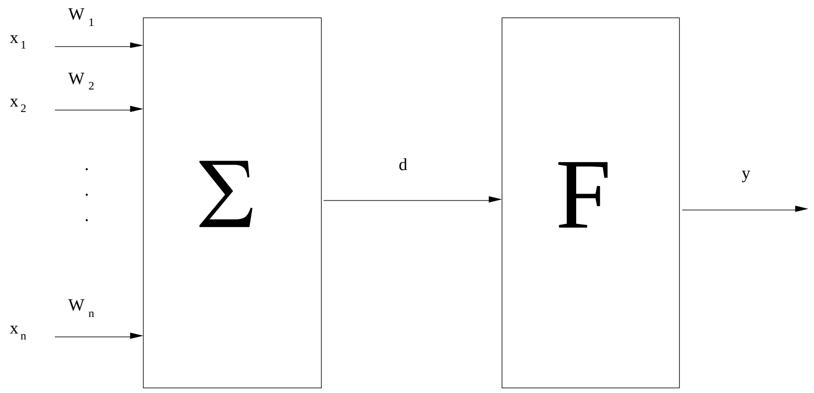
Сильний поштовх розвитку нейрокібернетики дав американський нейрофізіолог Френк Розенблатт, що запропонував в 1962 році свою модель нейронної мережі — персептрон. Сприйнятий спочатку з великим ентузіазмом, він незабаром піддався інтенсивним нападкам з боку знаних наукових авторитетів. І хоча докладний аналіз їх аргументів показує, що вони оспорювали не зовсім той персептрон, який пропонував Розенблатт, дослідження по нейронних мережах були згорнуті майже на 10 років.

Не дивлячись на це, в 70-і роки було запропоновано багато цікавих розробок, таких, наприклад, як когнитрон, здатний добре розпізнавати достатньо складні образи незалежно від повороту і зміни масштабу зображення. У київському інституті кібернетики з 70-х років ведуться роботи над стохастичними нейронними мережами. У 1982 році американський біофізик Дж. Хопфілд запропонував оригінальну модель нейронної мережі, названу його ім'ям. У подальші декілька років було знайдено безліч ефективних алгоритмів: мережа зустрічного потоку, двонаправлена асоціативна пам'ять і ін. У 1986 році Дж. Хінтон і його колеги опублікували статтю з описом моделі нейронної мережі і алгоритмом її навчання, що дало новий поштовх дослідженням в області штучних нейронних мереж.

І тільки в 90-х роках, коли обчислювальні системи стали достатньо потужними, НМ набули широкого поширення. Створення НМ було викликане спробами зрозуміти принципи роботи людського мозку і, без сумніву, це впливатиме і на подальший їх розвиток. Проте, порівняно з людським мозком, НМ сьогодні є вельми спрощеною моделлю. Не дивлячись на це, НМ вельми успішно використовуються при вирішенні самих різних завдань. Наприклад, програмна реалізація НМ може використовуватися для складання плану кредитних виплат людей, що звертаються в банк за позикою. Хоча і розв'язок на основі нейронних мереж може виглядати і поводитися як звичайне програмне забезпечення, вони **різні** в принципі, оскільки більшість реалізацій на основі нейронних мереж «навчаються», а не «програмуються»: мережа вчиться виконувати завдання, а не програмується безпосередньо. Насправді НМ використовуються тоді, коли неможливо написати відповідну програму, або внаслідок того знайденою НМ рішення виявляється досконалішим. Наприклад, експерт з продажу нерухомості зі свого досвіду може знати, які чинники впливають на продажну ціну кожного конкретного будинку, але при цьому часто є такі особливості, які буде вельми важко пояснити програмістові. Агентство з продажу нерухомості може побажати мати «провісника цін на основі НМ», навченого на безлічі прикладів реальних продажів тому, які чинники впливають на ціну об'єкту, що продається, і тому, яку відносну важливість має кожен з цих чинників. Але тут важливішим виявляється те, що рішення на основі НМ є гнучкішим, оскільки відповідна система може надалі удосконалювати точність прогнозів у міру накопичення нею досвіду і адаптуватися до змін, що відбуваються на ринку.

ШНМ є пристроєм паралельних обчислень, що складається сукупності простих елементів, з'єднаних та взаємодіючих між собою. Ці елементи, названі також ***нейронами*** або вузлами, є простими процесорами, обчислювальні можливості яких зазвичай обмежуються деяким правилом комбінування вхідних сигналів і правилом активізації, що дозволяє обчислити вихідний сигнал за сукупністю вхідних сигналів. Вихідний сигнал елементу може посилатися іншим елементам по зважених зв'язках, з кожним з яких пов'язаний ваговий коефіцієнт або вага. Залежно від значення вагового коефіцієнта передаваний сигнал або посилюється, або пригнічується.

Біологічний нейрон моделюється як пристрій, що має декілька входів (дендритів), і один вихід (аксон). Кожному входу ставиться у відповідність деякий ваговий коефіцієнт (w), що характеризує пропускну спроможність каналу і що оцінює ступінь впливу сигналу з цього входу на сигнал на виході. Залежно від конкретної реалізації, оброблювані нейроном сигнали можуть бути аналоговими або цифровими (1 або 0). У тілі нейрона відбувається зважене підсумовування вхідних збуджень, і далі це значення є аргументом активаційної функції нейрона, можливі з варіантів яких представлена на рис.1.



Сумуючий елемент $d = \sum W_i \cdot x_i$

Активізаційний елемент $y = F(d) = 1/\left(1 + e^{-d}\right)$

Рис. 1. Штучний нейрон

Структура зв'язків відображає деталі конструкції мережі, а саме те, які елементи сполучені, в якому напрямі працюють з'єднання і який рівень значущості (тобто вага) кожного з'єднання. Структура зв'язків зазвичай визначається в два етапи: спочатку розробник системи вказує, які елементи повинні бути зв'язані, і в якому напрямі, а потім в процесі фази навчання визначаються значення відповідних вагових коефіцієнтів.

Вагові коефіцієнти можна визначити і без проведення навчання, але якраз найбільша перевага НМ полягає в їх здатності навчатися виконанню завдання на основі тих даних, які мережа отримуватиме в процесі реальної роботи. Для багатьох застосувань навчання є не тільки засобом програмування мережі, коли немає достатніх знань про спосіб вирішення задачі, що дозволяють виконати програмування в традиційній формі, але часто єдиною метою навчання є перевірка того, що мережа дійсно зможе навчитися вирішувати поставлені перед нею завдання.

Незалежно від розташування та функціонального призначення, всі штучні нейронні елементи мають спільні компоненти.

Компонента 1. ***Вагові коефіцієнти.*** При функціонуванні нейрон одночасно отримує багато вхідних сигналів. Кожен вхід має свою власну синаптичну вагу, яка надає входу вплив, необхідний для функції суматора елемента обробки. Ваги є мірою сили вхідних зв'язків і моделюють різноманітні синаптичні сили біологічних нейронів. Ваги суттєвого входу підсилюються і навпаки вага несуттєвого входу примусово зменшується, що визначає інтенсивність вхідного сигналу. Ваги можуть змінюватись відповідно до навчальних прикладів, топології мережі та навчальних правил.

Компонента 2. *Функція суматора.* Першим кроком дії нейрону є обчислення зваженої суми всіх входів. Математично, вхідні сигнали та відповідні їм ваги представлені векторами (x_{10}, x_{20} ... x_{n0}) та (w_{10}, w_{20} . . . w_{n0}). Добуток цих векторів є загальним вхідним сигналом. Спрощеною функцією суматора є множення кожної компоненти вектора x на відповідну компоненту вектора w: вхід1 = $x_{10} * w_{10}$, вхід2 = $x_{20} * w_{20}$, і знаходження суми всіх добутків: вхід$_1$ + вхід$_2$ + . . . + вхід$_n$. Результатом є єдине число, а не багатоелементний вектор.

Функція суматора може бути складнішою, наприклад, вибір мінімуму, максимуму, середнього арифметичного, добутку або виконувати інший нормалізуючий алгоритм. Вхідні сигнали та вагові коефіцієнти можуть комбінуватись багатьма способами перед надходженням до передавальної функції. Особливі алгоритми для комбінування входів нейронів визначаються обраними мережною архітектурою та парадигмою.

В деяких нейромережах функції суматора виконують додаткову обробку, так звану функцію активації, яка зміщує вихід функції суматора відносно часу. На жаль, функції активації на теперішній час обмежено досліджені і більшість сучасних нейронних реалізацій використовують функцію активації "тотожності", яка еквівалентна її відсутності. Цю функцію доцільніше використовувати як компоненту мережі в цілому, ніж як компоненту окремого нейрона.

Компонента 3. *Передавальна функція.* Результат функції суматора є зваженою сумою вхідних сигналів, що перетворюється у вихідний сигнал через алгоритмічний процес відомий як передавальна функція. У передавальній функції для визначення виходу нейрона загальна сума порівнюється з деяким порогом. Якщо сума є більшою за значення порогу, елемент обробки генерує сигнал, в противному випадку сигнал не генерується або генерується гальмівний сигнал.

Переважно застосовують нелінійну передавальну функцію, оскільки лінійні (прямолінійні) функції обмежені і вихід є просто пропорційним до входу. Застосування лінійних передавальних функцій було проблемою у ранніх моделях мереж, і їх обмеженість та недоцільність була доведена в книзі Мінскі та Пейперта "Перцептрони".

На рис.2 зображені типові передавальні функції.

Для ***простої*** (рис. 2 а) передавальної функції нейромережа може видавати 0 та 1, 1 та -1 або інші числові комбінації. Передавальна функція в таких випадках є "жорстким обмежувачем" або пороговою функцією.

Інший тип передавальної функції - ***лінійна з насиченням*** (рис. 2 б) - віддзеркалює вхід всередині заданого діапазону і діє як жорсткий обмежувач за межами цього діапазону. Це лінійна функція, яка відсікається до мінімальних та максимальних значень, роблячи її нелінійною.

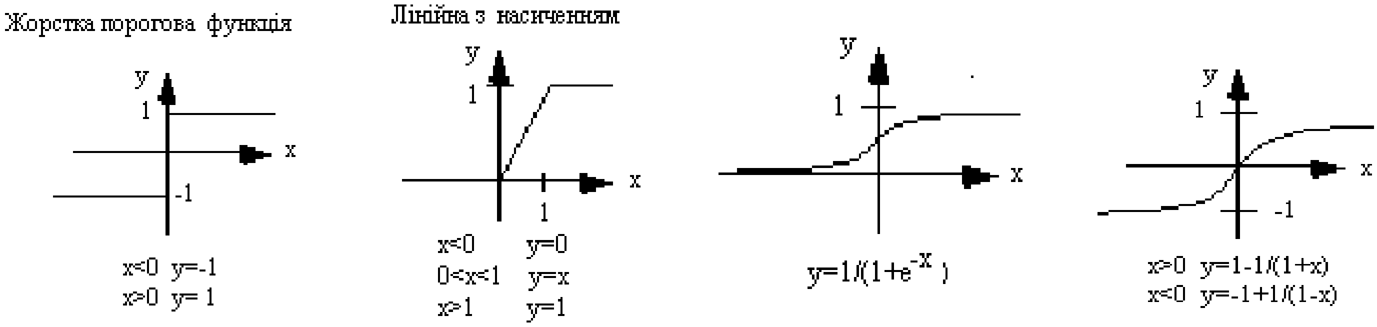


Рис. 2. Типові передавальні функції

Наступним вибором є ***сигмоїда*** (рис. 2 в) або S-подібна крива, яка наближує мінімальне та максимальне значення у асимптотах і називається сигмоїдою, коли її діапазон [0, 1], або ***гіперболічним тангенсом*** (рис. 2 г), при діапазоні [-1, 1]. Важливою рисою цих кривих є неперервність функцій та їх похідних. Застосування сигмоїдних функцій надає добрі результати і має широке застосування.

Зрештою, для різних нейромереж можуть вибиратись інші передавальні функції.

Перед надходженням до передавальної функції до вхідного сигналу деколи додають однорідно розподілений випадковий шум, джерело та кількість якого визначається режимом навчання. В літературі цей шум, згадується як "температура" штучних нейронів, яка надає математичній моделі елемент реальності.

Компонента 4. ***Масштабування.*** Після передавальної функції вихідний сигнал проходить додаткову обробку масштабування, тобто результат передавальної функції множиться на масштабуючий коефіцієнт і додається зміщення.

Компонента 5. ***Вихідна функція*** **(*змагання*)**. По аналогії з біологічним нейроном, кожний штучний нейрон має один вихідний сигнал, який передається до сотень інших нейронів. Переважно, вихід прямо пропорційний результату передавальної функції. В деяких мережних топологіях результати передавальної функції змінюються для створення змагання між сусідніми нейронами. Нейронам дозволяється змагатися між собою, блокуючи дії нейронів, що мають слабий сигнал. Змагання (конкуренція) може відбуватись між нейронами, які знаходяться на одному або різних прошарках. По-перше, конкуренція визначає, який штучний нейрон буде активним і забезпечить вихідний сигнал. По-друге, конкуруючі виходи допомагають визначити, який нейрон буде брати участь у процесі навчання.

Компонента 6. *Функція похибки та поширюване назад значення.* У більшості мереж, що застосовують контрольоване навчання обчислюється різниця між спродукованим та бажаним виходом. Похибка відхилення (поточна похибка) перетворюється функцією похибки відповідно заданій мережній архітектурі. В базових архітектурах похибка відхилення використовується безпосередньо, в деяких парадигмах використовується квадрат або куб похибки зі збереженням знаку.

Після проходження всіх прошарків поточна похибка поширюється назад до попереднього прошарку і може бути безпосередньо похибкою або похибкою, масштабованою певним чином залежно від типу мережі (наприклад похідною від передавальної функції). Це поширюване назад значення враховується в наступному циклі навчання.

Компонента 7. *Функція навчання*. Метою функції навчання є налаштування змінних ваг з'єднань на входах кожного елемента обробки відповідно до певного алгоритму навчання для досягнення бажаного результату. Існує два типи навчання: контрольоване та неконтрольоване. Контрольоване навчання вимагає навчальної множини даних або спостерігача, що ранжує ефективність результатів мережі. У випадку неконтрольованого навчання система самоорганізовується за внутрішнім критерієм, закладеним в алгоритм навчання.

3. Структура штучних нейронних мереж

Будучи сполученими певним чином, нейрони утворюють нейронну мережу. Робота мережі розділяється на навчання і адаптацію.

Адаптація – це процес зміни параметрів і структури системи, а можливо і керуючих дій, на основі поточної інформації з метою досягнення певного стану системи при початковій невизначеності і умовах роботи, що змінюються.

Навчання – це процес, в результаті якого система поступово набуває здатності відповідати потрібними реакціями на визначеній сукупності зовнішніх дій шляхом модифікації вагових коефіцієнтів зв'язків між нейронами. Цей процес є результатом алгоритму функціонування мережі, а не заздалегідь закладених в ній знань людини, як це часто буває в системах штучного інтелекту.

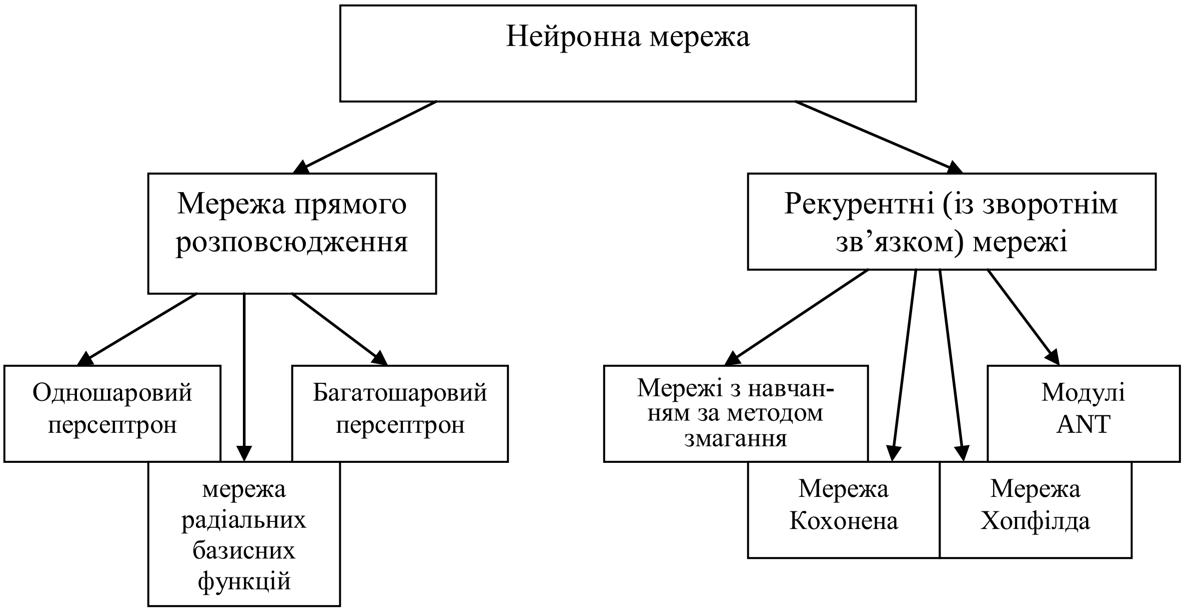


Рис. 3. Найбільш поширені штучні НМ

Серед різних структур НМ (рис. 3) однієї з найбільш відомих є багатошарова структура, в якій кожен нейрон довільного шару пов'язаний зі всіма аксонами нейронів попереднього шару або, у разі першого шару, зі всіма входами НМ. Такі НМ називаються повнозв'язними. Коли в мережі тільки один шар, алгоритм її навчання з вчителем досить очевидний, оскільки правильні вихідні стани нейронів єдиного шару свідомо відомі, і підстроювання синаптичних зв'язків йде в напрямі, що мінімізує помилку на виході мережі. За цим принципом будується, наприклад, алгоритм навчання одношарового перцептрона. У багатошарових же мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, невідомі, і два- або більше шаровий перцептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах НМ. Один з варіантів вирішення цієї проблеми – розробка наборів вихідних сигналів, відповідних вхідним, для кожного шару НМ, що, звичайно, є дуже трудомісткою операцією і не завжди здійсненною. Другий варіант – динамічне підлаштовування вагових коефіцієнтів синапсів, в ході якої вибираються, як правило, найбільш слабкі зв'язки і змінюються на малу величину в ту або іншу сторону, а зберігаються тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Очевидно, що даний метод "наукового тику", не дивлячись на свою простоту, вимагає громіздких рутинних обчислень. І, нарешті, третій, найприйнятніший варіант – розповсюдження сигналів помилки від виходів НМ до її входів, в напрямі, зворотному прямому розповсюдженню сигналів в звичайному режимі роботи (алгоритм зворотного розповсюдження). Всі НМ можна представити за допомогою наступних абстракцій:

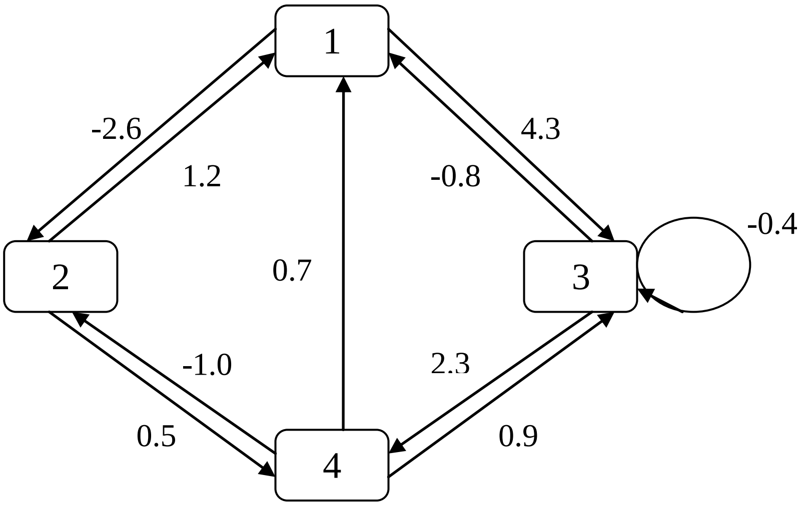
- Множина простих процесорів
- Структура зв'язків
- Правило розповсюдження сигналів в мережі
- Правило комбінування вхідних сигналів
- Правило обчислення сигналу активності
- Правило навчання, що коректує зв'язки.

Множина простих процесорів.

З кожним процесором, тобто оброблювальним елементом мережі, зв'язується набір ***вхідних зв'язків***, по яких до даного елементу поступають сигнали від інших елементів мережі, і набір ***вихідних зв'язків***, по яких сигнали даного елементу передаються іншим елементам. Деякі елементи призначені для отримання сигналів із зовнішнього середовища (і тому називаються ***вхідними елементами***), а деякі – для виводу в зовнішнє середовище результатів обчислень (***вихідні елементи***).

Структура зв'язків.

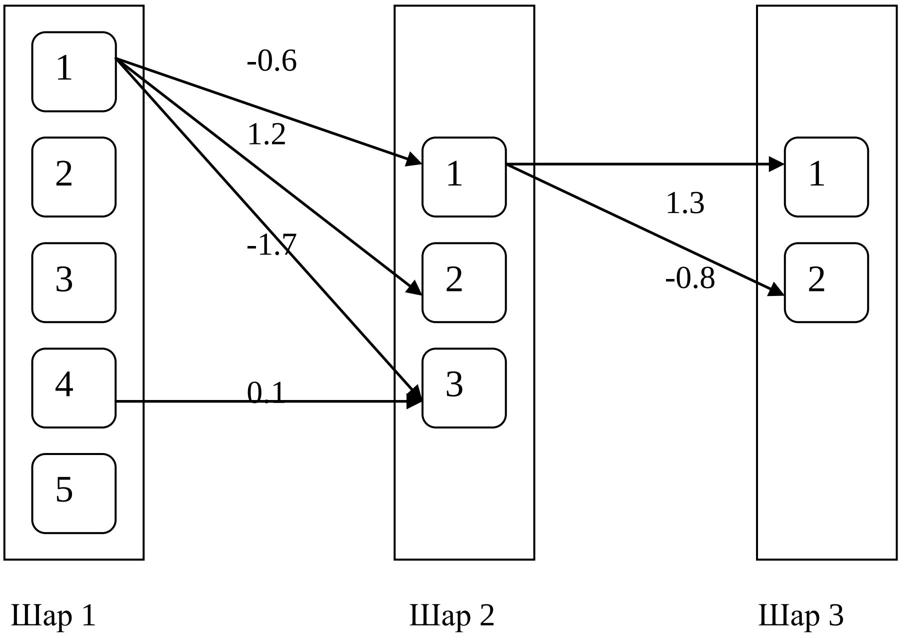
Структура зв'язків відображає, як сполучені елементи мережі між собою. У одній моделі (тобто для одного типу мережі) кожен елемент може бути пов'язаний зі всіма іншими елементами мережі, в іншій моделі елементи можуть бути організовані в деякій впорядкованій за рівнями (шарах) ієрархії, де зв'язки допускаються тільки між елементами в суміжних шарах, а в третій – можуть допускатися зворотні зв'язки між суміжними шарами або всередині одного шару, або ж допускатися посилання сигналів елементами самим собі. Зазвичай для кожної конкретної моделі мережі вказується тип допустимих зв'язків. ***Кожен зв'язок визначається трьома параметрами***: елементом, від якого виходить даний зв'язок, елементом, до якого даний зв'язок направлений, і числом, що вказує ваговий коефіцієнт. Негативне значення ваги відповідає придушенню активності відповідного елементу, а позитивною значення – посиленню його активності. Абсолютне значення вагового коефіцієнта характеризує силу зв'язку.



$$w = \begin{bmatrix} 0.0 & -2.6 & 4.3 & 0.0 \\ 1.2 & 0.0 & 0.0 & 0.5 \\ -0.8 & 0.0 & -0.4 & 0.9 \\ 0.7 & -1.0 & 2.3 & 0.0 \end{bmatrix}$$

Рис. 4. Представлення структури зв'язків одношарової ШНМ

Структура зв'язків зазвичай представляється у вигляді вагової матриці *W*, в якій кожен елемент w_{ij} представляє величину вагового коефіцієнта для зв'язку, що йде від елементу *i* до елементу *j*. Якщо елементи мережі описуються не однією, а декількома ваговими матрицями, тоді елементи мережі виявляються згрупованими в шари. На рис. 4 і 5 пропонуються приклади представлення структури зв'язків у вигляді відповідних матриць.

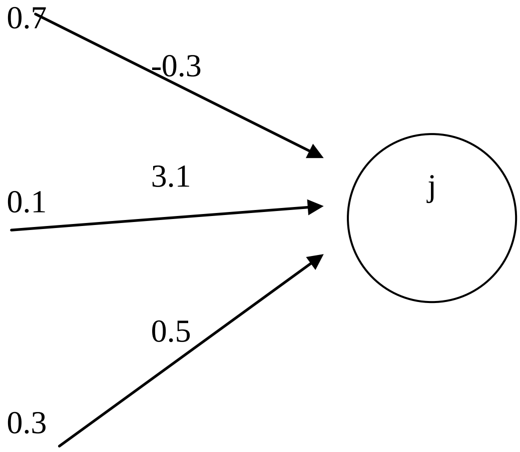


$$W_1 = \begin{bmatrix} -0.6 & 1.2 & -1.7 \\ . & . & . \\ . & . & . \\ . & . & 0.1 \\ . & . & . \end{bmatrix} \quad W_2 = \begin{bmatrix} 1.3 & -0.8 \\ . & . \\ . & . \end{bmatrix}$$

Рис. 5. Представлення структури зв'язків двошарової ШНМ

Правило розповсюдження сигналів в мережі.

Досить часто вхідні сигнали елементу передбачається комбінувати шляхом підсумовування їх зважених значень. Приклад цього методу підсумовування показаний на рис. 6, де net_j позначає результат комбінування введення елементу *j*, x_i – вихід елементу *i*, а *n* – число задіяних зв'язків.



$$\boldsymbol{net_j = \sum_{i=1}^{n} x_i w_i}$$

$$\boldsymbol{net_j} = (0.7 \times -0.3) + (0.1 \times 3.1) + (0.3 \times 0.5) = 0.25$$

або у векторному представленні

$$\begin{bmatrix} 0.7 & 0.1 & 0.3 \end{bmatrix} \begin{bmatrix} -0{,}3 \\ 3.1 \\ 0.5 \end{bmatrix}$$

Рис. 6. Розповсюдження сигналів в мережі

Використовуються і інші форми комбінування вхідних сигналів. Іншим методом, що часто зустрічається, є квадрат різниці між значенням сили зв'язку і значенням передаваного по зв'язку сигналу з подальшим підсумовуванням таких різниць для всіх вхідних зв'язків даного елементу.

Правило обчислення сигналу активності.

Для всіх елементів є правило обчислення вихідного значення, яке передбачається передати іншим елементам або в зовнішнє середовище. Це правило називають правилом активності, а відповідне вихідне значення називають активністю відповідного елементу. Активність може представлятися або деяким дійсним значенням довільного вигляду, або дійсним значенням з деякого обмеженого інтервалу значень (наприклад, з інтервалу [0,1]), або ж деяким значенням з певного дискретного набору значень (наприклад, $\{0,1\}$ або)$\{+1,-1\}$. На вхід функції активності поступає значення комбінованого введення даного елементу. Приклади функцій активності наводяться нижче.

Правило обчислення сигналу активності – тотожна функція.

Функція активності для вхідних елементів може бути тотожною функцією, і це просто означає, що значення (сигнал, що посилається іншим елементам) опиняється в точності рівним комбінованому введенню (рис. 7). Вхідні елементи зазвичай призначені для розподілу сигналів, що вводяться, між іншими елементами мережі, тому для вхідних елементів зазвичай потрібно, щоб вихідний від елементу сигнал був таким же, як і вхідний. На відміну від інших елементів мережі, вхідні елементи мережі мають тільки одне вхідне значення. Наприклад, кожен вхідний елемент може отримувати сигнал від одного відповідного йому давача, розміщеного на фюзеляжі літака. Один цей елемент зв'язується з багатьма іншими елементами мережі, так що дані, отримані від одного давача, виявляються розподіленими між багатьма елементами мережі.

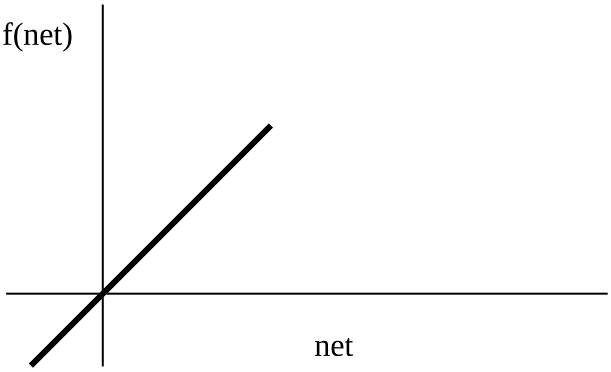
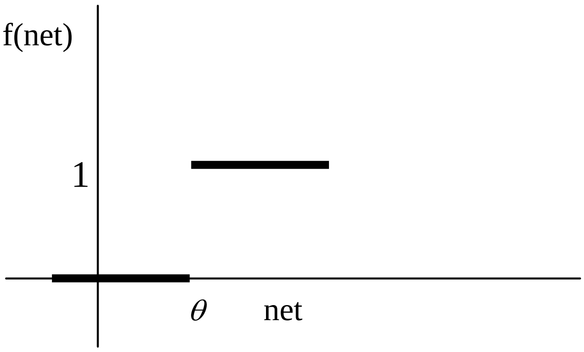


Рис. 7.

Правило обчислення сигналу активності – порогова функція.

У більшості моделей НМ використовуються нелінійні функції активності. Порогова функція обмежує активність значеннями 1 або 0 залежно від значення комбінованого введення порівняно з деякою пороговою величиною θ (рис. 8).



$$f(net) = \begin{cases} 1, & \text{якщо} \quad net_i \geq \theta \\ 0, & \text{якщо} \quad net_i \leq \theta \end{cases}$$

Рис. 8. Порогова функція

Найчастіше зручніше відняти порогове значення (зване зсувом або зрушенням) із значення комбінованого введення і розглянути порогову функцію в її математично еквівалентній формі, показаній на рис. 9. Зрушення w_0 в даному випадку виявляється негативним, а значення комбінованого введення обчислюється за формулою

$$f(net) = \begin{cases} 1, & \text{якщо} \quad net_i \geq 0 \\ 0, & \text{якщо} \quad net_i \leq 0 \end{cases}, \quad net_j = w_o + \sum_{i=1}^{n} x_i w_i \, .$$

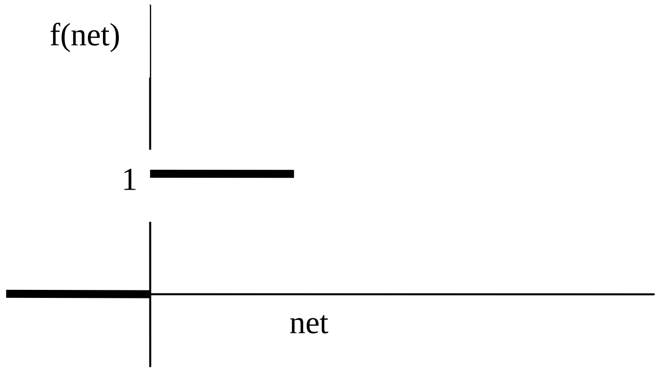


Рис. 9. Порогова функція без зсуву

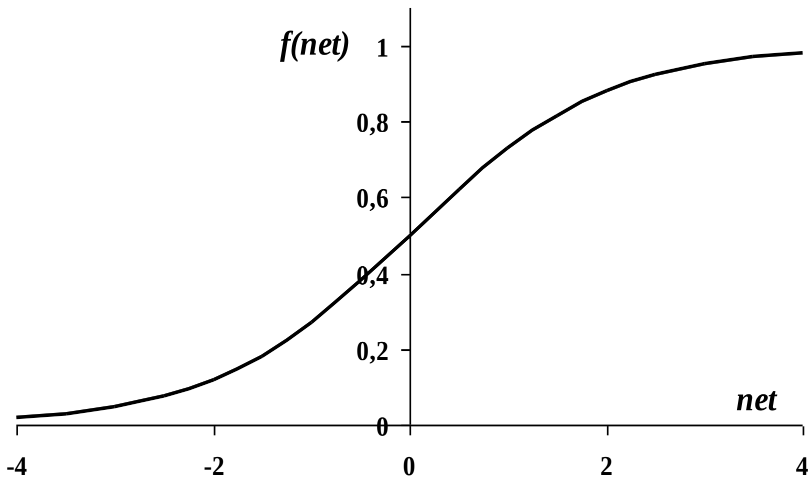
Зрушення зазвичай інтерпретується як зв'язок. Витікаюча від елементу активність якого завжди рівна 1. комбіноване введення в цьому випадку можна представити у вигляді

$$net_j = \sum_{i=0}^{n} x_i w_i$$

де x_0 завжди вважається рівним 1.

Правило обчислення сигналу активності – сигмоїдна функція.

Найбільш часто використовуваною функцією активності є сигмоїдна функція. Вихідні значення такої функції безперервно заповнюють діапазон від 0 до 1. прикладом може служити логістична функція, показана на рис. 10.



$$f(net) = \frac{1}{1 + exp(-net)}$$

Рис. 10. Сигмоїдна порогова функція

Нахил і область вихідних значень логістичної функції можуть бути різними. Наприклад, для біполярного сигмоїда областю вихідних значень є діапазон -1 і 1.

В нейронних мережах застосовуються кілька варіантів сигмоїдних передавальних функцій.

Функція Ферми (експонентна сигмоїда):

$$f(S) = \frac{1}{1 + e^{-2\alpha S}}$$

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

Раціональна сигмоїда:

$$f(S) = \frac{S}{|S| + \alpha}$$

Гіперболічний тангенс:

$$f(S) = th\frac{S}{\alpha} = \frac{e^{-\frac{S}{\alpha}} - e^{-\frac{S}{\alpha}}}{e^{\frac{S}{\alpha}} + e^{-\frac{S}{\alpha}}}$$

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

$$f(S) = p_1 \frac{S}{|S| + p_2} + p_3$$

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

Правило навчання, що коректує зв'язки

Якість роботи НМ сильно залежить від того, що пред'являється їй в процесі набору учбових даних. Учбові дані повинні бути типовими для завдання, вирішенню якого навчається мережа. Навчання часто виявляється унікальним процесом, коли прийнятні вирішення багатьох проблем можуть бути отримані тільки в процесі численних експериментів. Розробникам рішення на основі нейронної мережі потрібне наступне.

- Вибрати відповідну модель мережі.
- Визначити топологію мережі (тобто число елементів і їх зв'язку)
- Вказати параметри навчання.

Існують ***три парадигми навчання***: з вчителем, без вчителя і змішана.

4. Контрольоване навчання (навчання зі вчителем)

Навчання з вчителем припускає, що для кожного вхідного вектора існує цільовий вектор, що є необхідним виходом (рішення задачі). Разом він називаються навчальною парою. Зазвичай мережа навчається на деякому числі таких навчальних пар.

Коли в мережі один шар, алгоритм її навчання з вчителем досить очевидний, оскільки правильні вихідні стани нейронів єдиного шару свідомо відомі, і підстроювання синаптичних зв'язків йде в напрямі, що мінімізує помилку на виході мережі.

У багатошарових мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, не відомі, і дво- або більш шарову НМ вже неможливо навчити, керуючись тільки величинами помилок на виходах мережі. Один з варіантів вирішення цієї проблеми – розробка наборів вихідних сигналів, відповідних вхідним, для кожного шару НМ, що звичайно є трудомісткою операцією і не завжди здійсненно. Другий варіант – динамічне підстроювання вагових коефіцієнтів синапсів, в ході якої вибираються найбільш слабкі зв'язки, які змінюються на малу величину в ту або іншу сторону. Зберігаються ж тільки ті зміни, які спричинили зменшення помилки на виході мережі. Очевидно, що даний метод «проб» вимагає громіздких рутинних обчислень.

Третій, найбільш прийнятний варіант – розповсюдження сигналів помилки від виходів НМ до її входів, в напрямі зворотному прямому розповсюдженню сигналів в звичайному режимі роботи. Цей алгоритм навчання НМ отримав назву процедури зворотного розповсюдження і є найбільш широко поширеним.

Повний алгоритм навчання НМ за допомогою процедури зворотного розповсюдження будується так:

1. Подати на входи мережі один з можливих образів і в режимі звичайного функціонування НМ, коли сигнали розповсюджуються від входів до виходів, розрахувати значення останніх за формулою

$$s_j^{(n)} = \sum_{i=0}^{M} y_i^{(n-1)} \cdot w_{ij}^{(n)} \qquad (1)$$

де: M – число нейронів в шарі n-1 зі врахуванням нейрона з постійним вихідним станом +1, який задає зсув;

s_j – зважена сума вхідних сигналів нейрона, тобто аргумент активаційної функції;

$y_i^{(n-1)}=x_{ij(n)}$ – i-ий вхід нейрона j шару n.

y_j – вихід нейрона j;

w_{ij} – ваговий коефіцієнт синаптичного зв'язку, що з'єднує i-ий нейрон шару n-1 з j-им нейроном шару n.

Мережі на кроці 1 попеременно у випадковому порядку задаються всі тренувальні образи, щоб мережа, образно кажучи, не забувала одні по мірі запам'ятовування інших.

2. Розрахувати $\delta^{(N)}$ для вихідного шару за формулою

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l}, \qquad (2)$$

де: $y_{j,p}^{(N)}$ – реальний вихідний стан нейрона j вихідного шару N нейронної мережі при подачі на її входи p-го образу;

d_{jp} – ідеальний (бажаний) вихідний стан цього нейрона.

Оскільки множник dy_j/ds_j є похідною активаційної функції по її аргументу, з цього виходить, що похідна цієї функції повинна бути визначена на всій осі абсцис. У зв'язку з цим функція одиничного стрибка і інші активаційні функції з неоднорідностями не підходять для даних НМ. У них застосовуються такі гладкі функції, як гіперболічний тангенс або класичний сигмоїд з експонентою. У випадку гіперболічного тангенса

$$\frac{dy}{ds} = 1 - s^2 .$$

Розрахувати зміни ваг w(N) шару N за формулами:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \qquad (9)$$

або

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \qquad (10)$$

де μ – коефіцієнт інерційності,

t – номер поточної ітерації;

η – коефіцієнт швидкості навчання, 0<<1.

3. Розрахувати (n)δ і ***w(n)*** для решти всіх шарів***, n=N-1...1*** за формулами відповідно:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)}\right] \cdot \frac{dy_j}{ds_j}, \quad \Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}$$

або

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j}, \quad \Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) .$$

4. Скоректувати всі ваги в НМ

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \qquad (14)$$

5. Якщо помилка мережі істотна, перейти на крок 1. Інакше – кінець.

Згідно методу найменших квадратів, цільовою функцією помилки НМ, що мінімізується, є величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2 \qquad (1)$$

Сумування ведеться по всіх нейронах вихідного шару і по всіх оброблюваних мережею образах. Мінімізація ведеться методом градієнтного спуску, що означає налаштовування вагових коефіцієнтів таким чином:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \qquad (2)$$

З виразу (9) виходить, що коли вихідне значення $yi^{(n-1)}$ прямує до нуля, ефективність навчання помітно знижується. При двійкових вхідних векторах в середньому половина вагових коефіцієнтів не коректуватиметься, тому область можливих значень виходів нейронів [0,1] бажано вибрати в межах [-0.5,+0.5], що досягається простими модифікаціями логістичних функцій. Наприклад, сигмоїд з експонентою перетвориться до вигляду

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha \cdot x}} \qquad (15)$$

5. Неконтрольоване навчання (навчання без вчителя)

Неконтрольоване навчання може бути великим надбанням у майбутньому. Воно проголошує, що комп'ютери можуть самонавчатись у справжньому роботизованому сенсі. На даний час, неконтрольоване навчання використовується мережах, відомих як самоорганізовані карти (*self organizing maps*), що знаходяться в досить обмеженому користуванні, але доводять перспективність самоконтрольованого навчання.

Мережі не використовують зовнішніх впливів для коректування своїх ваг і внутрішньо контролюють свою ефективність, шукаючи регулярність або тенденції у вхідних сигналах та роблять адаптацію згідно навчальної функції. Навіть без повідомлення правильності чи неправильності дій, мережа повинна мати інформацію відносно власної організації, яка закладена у топологію мережі та навчальні правила.

Алгоритм неконтрольованого навчання скерований на знаходження близькості між групами нейронів, які працюють разом. Якщо зовнішній сигнал активує будь-який вузол в групі нейронів, дія всієї групи в цілому збільшується. Аналогічно, якщо зовнішній сигнал в групі зменшується, це приводить до гальмуючого ефекту на всю групу.

Конкуренція між нейронами формує основу для навчання. Навчання конкуруючих нейронів підсилює відгуки певних груп на певні сигнали. Це пов'язує групи між собою та відгуком. При конкуренції змінюються ваги лише нейрона-переможця.

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.