

Тема 9, частина 2: Нейронні мережі

План

1. Поняття та можливості нейрокомп'ютерних технологій.
2. Архітектура нейронних мереж.
3. Навчання нейромережі. Зворотнє поширення похибки.

1. Поняття та можливості нейрокомп'ютерних технологій

Нейронні мережі - це адаптивні системи для обробки та інтелектуального аналізу даних, які є математичною структурою, що імітує деякі аспекти роботи людського мозку і демонструє такі його можливості, як здібність до неформального навчання, узагальнення і кластеризації некласифікованої інформації, здатність самостійно будувати прогнози на основі спостереження часових рядів. Головною їх відмінністю від інших методів є те, що нейромережі в принципі не потребують заздалегідь відомої моделі, а будують її самі лише на основі інформації, яку отримали. Саме тому нейронні мережі увійшли до практики усюди, де потрібно вирішувати задачі прогнозування, класифікації, управління - іншими словами, в області людської діяльності, де є задачі, що погано алгоритмізуються, для вирішення яких необхідні або постійна робота групи кваліфікованих експертів, або адаптивні системи автоматизації, якими і є нейронні мережі.

Нейронна мережа приймає вхідну інформацію і аналізує її способом, аналогічним тому, що використовує наш мозок. Під час аналізу мережа вивчає (набуває досвіду і знання) і видає вихідну інформацію на основі придбаного раніше досвіду. Основне завдання аналітика, що використовує нейронні мережі для вирішення якої-небудь проблеми, - створити найбільш ефективну архітектуру нейронної мережі, тобто правильно вибрати вигляд нейронної мережі, алгоритм її навчання, кількість нейронів і види зв'язків між ними. Ця робота не має формалізованих процедур, вона вимагає глибокого розуміння різних видів архітектури нейронних мереж, включає багато дослідницької і аналітичної роботи, і може зайняти досить багато часу.

Для неформалізованих завдань нейромережеві моделі можуть на порядок перевершувати традиційні методи рішення. Але використання нейронних мереж доцільне, якщо:

- накопичені достатні об'єми даних про попередню поведінку системи;
- не існує традиційних методів або алгоритмів, що задовільно вирішують проблему;

- дані частково спотворені, частково суперечливі або не повні і тому традиційні методи видають незадовільний результат.

Нейронні мережі найкраще проявляють себе там, де є велика кількість вхідних даних, між якими існують неявні взаємозв'язки і закономірності. В цьому випадку нейромережі допоможуть автоматично врахувати різні нелінійні залежності, приховані в даних. Це особливо важливо в системах підтримки прийняття рішень і системах прогнозування. Нейромережі є незамінними при аналізі даних, зокрема, для попереднього аналізу або відбору, виявлення «випадних фактів» або грубих помилок людини, що приймає рішення. Доцільно використовувати нейромережеві методи в задачах з неповною або «зашумленою» інформацією, особливо в задачах, де рішення можна знайти інтуїтивно, і при цьому традиційні математичні моделі не дають бажаного результату.

Методи нейронних мереж можуть використовуватися незалежно або ж служити прекрасним доповненням до традиційних методів статистичного аналізу, більшість з яких пов'язані з побудовою моделей, заснованих на тих або інших припущеннях і теоретичних висновках (наприклад, що досліджуєма залежність є лінійною або що деяка змінна має нормальний розподіл). Нейромережевий підхід не пов'язаний з такими припущеннями - він однаково придатний для лінійних і складних нелінійних залежностей, особливо ж ефективний в розвідувальному аналізі даних, коли ставиться мета з'ясувати, чи є залежності між змінними. При цьому дані можуть бути неповними, суперечливими і навіть свідомо спотвореними. Якщо між вхідними і вихідними даними існує якийсь зв'язок, що навіть не виявляється традиційними кореляційними методами, то нейронна мережа здатна автоматично налаштуватися на його із заданою мірою точності. Крім того, сучасні нейронні мережі володіють додатковими можливостями: вони дозволяють оцінювати порівняльну важливість різних видів вхідної інформації, зменшувати її об'єм без втрати істотних даних, розпізнавати симптоми наближення критичних ситуацій і так далі.

Штучні нейронні мережі породжені біологією, оскільки вони складаються з елементів, функціональні можливості яких аналогічні більшості елементарних функцій біологічного нейрона. Ці елементи потім організуються за способом, який може відповідати (або не відповідати) анатомії мозку. Не дивлячись на таку поверхневу схожість, штучні нейронні мережі демонструють деякі властивості мозку. Наприклад, вони вивчаються на основі досвіду, узагальнюють попередні прецеденти на нові випадки і витягують істотні властивості з інформації, що поступає та містить зайві дані.

<p>Традиційні ЕОМ</p> <ul style="list-style-type: none"> • Послідовні • Заданий алгоритм • Ієрархична структура алгоритмів та розподіл складної задачі на декілька простих 	<p>Нейрокомп'ютери</p> <ul style="list-style-type: none"> • Паралельні • Алгоритм формується шляхом навчання • Операції з образами
---	---

Серед головних проблем, які вирішують за допомогою штучних НМ і які цікавлять вчених та інженерів, можна визначити такі:

- класифікація образів: мета задачі – вказати приналежність вхідного образу (наприклад, речового сигналу або рукописного символу), поданого як вектор ознак, до одного або кількох завчасно визначених класів. До таких застосувань належать розпізнавання букв, мови, класифікація сигналу електрокардіограми, кліток крові тощо;
- кластеризація / категоризація: при розв'язанні задач кластеризації, званої як класифікація образів «без вчителя», відсутня навчальна вибірка з позначками класів. Алгоритм кластеризації будують на основі образів і розташовують подібні образи в один кластер. Відомі випадки застосування кластеризації для здобуття знань, скорочення обсягу даних і дослідження їх властивостей;
- апроксимація функцій: припустимо, що існує вибірка $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ (пари даних вхід–вихід), що генерується відомою функцією $F(x)$, котра спотворюється через шум. Задача апроксимації зводиться до знаходження оцінки невідомої функції $F_n(x)$. Апроксимація функцій необхідна при розв'язанні численних інженерних задач моделювання;
- передбачення / прогноз: нехай задані n дискретних точок $\{y(t_1), y(t_2), \dots, y(t_n)\}$ у послідовні проміжки часу t_1, t_2, \dots, t_n . У задачі потрібно передбачити значення $y(t_{n+1})$ у певний майбутній момент часу t_{n+1} . Передбачення / прогноз має значний вплив на прийняття рішень у бізнесі, науці та техніці. Передбачення цін на фондовій біржі та прогноз погоди – це типові застосування технології передбачення / прогнозу;
- оптимізація: численні проблеми в математиці, статистиці, техніці, науці, медицині й економіці можна розглядати як проблеми оптимізації. Задача алгоритма оптимізації – знаходження такого рішення, що задовольняє систему обмежень і максимізує або мінімізує цільову функцію. Задача комівояжера з класу NP-повних – це класичний приклад задачі оптимізації;
- асоціативна пам'ять: у моделі розрахунків фон Неймана звернення до пам'яті можливе лише за допомогою адреси, що не залежить від змісту

пам'яті. Крім того, якщо допущена помилка в розрахунку адреси, то можна знайти зовсім іншу інформацію. Асоціативна пам'ять (пам'ять, що адресується за змістом) доступна після визначення заданого змісту. Вміст пам'яті можна викликати навіть у результаті часткового входження або за спотвореним змістом. Асоціативна пам'ять надзвичайно важлива при створенні мультимедійних інформаційних баз даних;

- управління: розглянемо динамічну систему, задану сукупністю $\{u(t), y(t)\}$, де $u(t)$ – вхідний керуючий вплив, а $y(t)$ – вихід системи в момент часу t . У системах управління з еталонною моделлю головна мета управління – це розрахунок такого вхідного впливу $u(t)$, при якому система прямує по бажаній траєкторії, що диктує еталонна модель. Прикладом є оптимальне управління двигуном.

2. Архітектура нейронних мереж.

Означення. Штучні нейронні мережі - вид математичних моделей, які будуються за принципом організації і функціонування їх біологічних аналогів, - мереж нервових клітин (нейронів) мозку. У основі їх побудови лежить ідея про те, що нейрони можна моделювати досить простими автоматами (штучними нейронами), а вся складність мозку, гнучкість його функціонування і інші найважливіші якості визначаються зв'язками між нейронами.

Розглянемо математичну модель штучного нейрону (рис. 9.3).

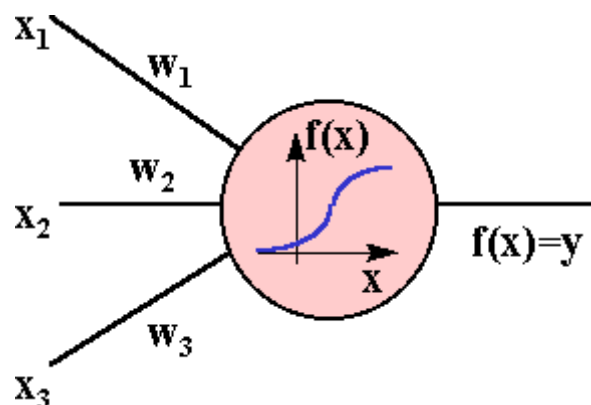


Рис. 9.3. Модель нейрона.

На рисунку зображена модель нейрона з трьома входами (дендритами), причому синапси цих дендритів мають ваги w_1 , w_2 , w_3 . Хай до синапсів поступають імпульси сили x_1 , x_2 , x_3 відповідно, тоді після проходження синапсів і дендритів до нейрона поступають імпульси $w_1 x_1$, $w_2 x_2$, $w_3 x_3$. Нейрон перетворює отриманий сумарний імпульс $x = w_1 x_1 + w_2 x_2 + w_3 x_3$

відповідно до деякої передавальної функції $f(x)$. Сила вихідного імпульсу дорівнює $y = f(x) = f(w_1 x_1 + w_2 x_2 + w_3 x_3)$. Таким чином, нейрон повністю описується своїми вагами w_k і передавальною функцією $f(x)$. Отримавши набір чисел (вектор) x_k як входи, нейрон видає деяке число y на виході.

Нелінійна функція $f(x)$ також називається активаційною і може мати різний вигляд, як показано на рисунку 9.4. Однією з найбільш поширених є нелінійна функція з насиченням, так звана логістична функція або сигмоїд (функція S-образного вигляду):

$$f(x) = \frac{1}{1 + e^{-ax}}$$

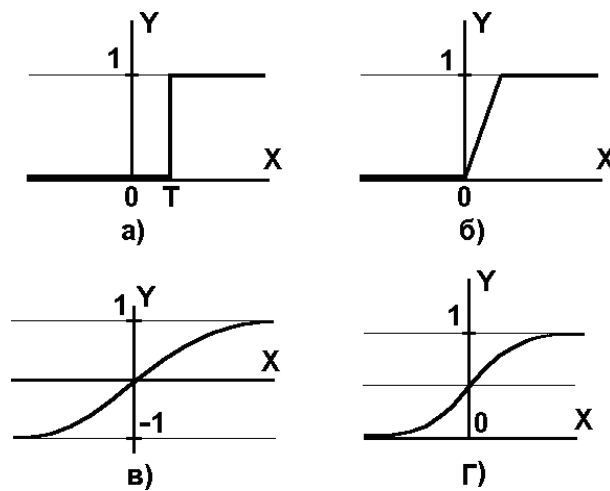


Рис. 9.4. Вигляд активаційної функції: а) функція одиничного стрибка; б) лінійний поріг (гистерезис); в) сигмоїд – гіперболічний тангенс; г) сигмоїд – функція з насиченням.

З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить в діапазоні $[0,1]$.

Уявімо собі найпростішу мережу – так званий одношаровий перцептрон. Він складається з одного шару нейронів.

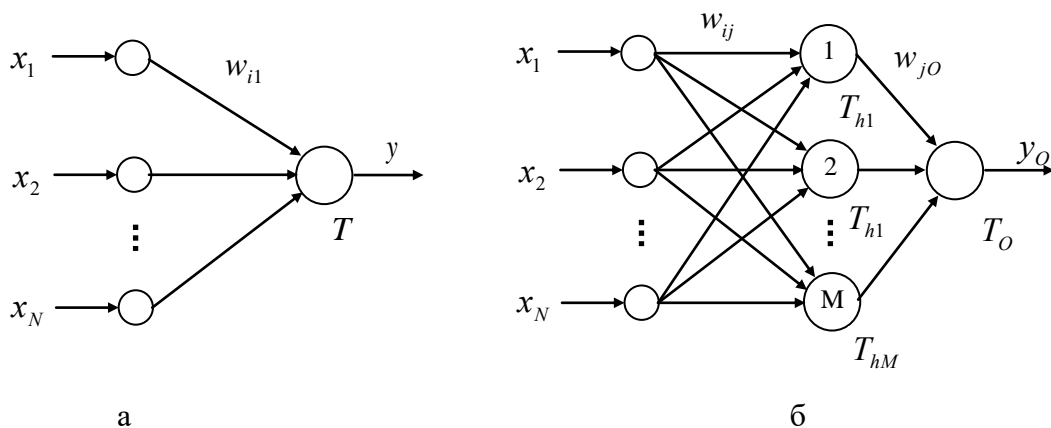


Рис. 9.5. Структура НМ: а – однорівневий перцептрон; б – багаторівневий перцептрон.

Його архітектура виглядає так: є певна кількість сенсорних (вхідних) нейронів, кожен з яких може активуватися в певному діапазоні залежно від архітектури (0, 1 або -1, 1 або -1 тощо). Нейрон має зв'язки з іншими нейронами на попередньому шарі. У випадку одношарового перцептрона – це зв'язок із даними, які ми подаємо на вхід.

Коли нейрон отримав вхідний імпульс, він має зв'язки із іншими нейронами і, залежно від того, яка сила цих зв'язків, він впливає на активацію інших нейронів.

Вони, в свою чергу, отримуючи імпульси з попереднього шару, активуються на певний рівень, залежно від сили зв'язків, і передають естафету наступному шару.

Один нейрон окремо – це проста система, яка множить на певний набір коефіцієнтів вхідні імпульси, сумує і застосовує певну функцію активації (лінійну, сигмоїдальну, обмежену в певних діапазонів).

Функція активації – це спосіб трансформації нашого імпульсу в іншу форму. Наприклад, якщо нам потрібно зробити, щоб нейрон завжди активувався в межах від 0 до 1 незалежно від сили імпульсу від попередніх нейронів, ми застосовуємо сигмоїдальну функцію активації.

Отже, один окремий нейрон є досить примітивною і простою математичною функцією. Але ще в середині 20 століття стало зрозуміло, що системи з великою кількістю нейронів можуть апроксимувати (наближати і описувати) досить непрості закономірності, знаходити закономірності між вхідними і вихідними даними і будувати моделі-узагальнення. Тобто НМ чудово підходять для задач моделювання певних даних і прогнозування поведінки цих даних на тестовій вибірці, а не тільки на тренувальній вибірці.

НМ – це математична модель, яка складається із нейронів – маленьких апроксиматорів – які мають зв'язки із попереднім шаром. Іншими словами, нейронна мережа – це певна функція, яка має величезний набір параметрів, які ми даємо на вхід (т.зв. вектор ознак). Ця функція перемножує велику кількість разів

параметри на певні навчені коефіцієнти таким чином, що на виході ми отримуємо і аналізуємо інший вектор необхідної розмірності.

В нейронних мережах завжди на вхід подається певний вектор і на виході теж отримується вектор. Далі цей вектор можна проаналізувати. Система машинного навчання має справу тільки з векторами в тому чи іншому виді і деформує їх в багатовимірному просторі.

У зв'язку з великою різноманітністю штучних нейронних мереж і специфікою їх реалізації важливим є проведення класифікацію їх архітектур для подальшого застосування в інтелектуальних системах аналізу даних.

- Нейронні мережі можуть бути *синхронні і асинхронні*.

У синхронних нейронних мережах в кожен момент часу свій стан міняє лише один нейрон. У асинхронних - стан міняється відразу у цілої групи нейронів, як правило, у всього шару.

- По типу базової архітектури - *шаровані і повнозв'язні мережі*.

Ключовим в шаруватих мережах є поняття прошарка. Шар - один або декілька нейронів, на входи яких подається один і той же загальний сигнал. Шаровані нейронні мережі - нейронні мережі, в яких нейрони розбиті на окремі групи (шари) так, що обробка інформації здійснюється пошарово. У шарованих мережах нейрони і-го шару отримують вхідні сигнали, перетворюють їх і через точки розгалуження передають нейронам (i+1) шару. І так до k-го шару, який видає вихідні сигнали для інтерпретатора і користувача. Число нейронів в кожному шарі не пов'язане з кількістю нейронів в інших шарах і може бути довільним. В рамках одного шару дані обробляються паралельно, а в масштабах всієї мережі обробка ведеться послідовно - від шару до шару.

Шаровані мережі, у свою чергу, можуть бути *одношаровими і багатошаровими*. Одношарова мережа – мережа складається з одного шару. Багатошарова мережа – мережа має декілька шарів. У багатошаровій мережі перший шар називається вхідним, подальші - внутрішніми або прихованими, останній шар - вихідним. Таким чином, проміжні шари - це всі шари в багатошаровій нейронній мережі, окрім вхідного і вихідного. Вхідний шар мережі реалізує зв'язок з вхідними даними, вихідний - з вихідними.

Таким чином, нейрони можуть бути вхідними, вихідними і прихованими. Вхідний шар організований з вхідних нейронів (input neuron), які отримують дані і поширюють їх на входи нейронів прихованого шару мережі. Прихований нейрон (hidden neuron) - це нейрон, що знаходиться в прихованому шарі нейронної мережі. Вихідні нейрони (output neuron), з яких організований вихідний шар мережі, видає результати роботи нейронної мережі.

У повнозв'язних мережах кожен нейрон передає свій вихідний сигнал решті нейронів, включаючи самого себе. Вихідними сигналами мережі можуть бути все

або деякі вихідні сигнали нейронів після декількох тактів функціонування мережі. Всі вхідні сигнали подаються всім нейронам.

- По спрямованості зв'язків.

Нейронні мережі бувають *із зворотними зв'язками і без зворотних зв'язків*.

Серед мереж без зворотних зв'язків розрізняють *мережі із зворотним розповсюдженням помилки та інші мережі*. Мережі першої групи характеризуються фіксованою структурою, ітераційним навчанням, корегуванням вагів по помилках. Перевагами мереж без зворотних зв'язків є простота їх реалізації і гарантоване отримання відповіді після проходження даних по шарах. Недоліком цього виду мереж вважається мінімізація розмірів мережі - нейрони багато разів беруть участь в обробці даних. Менший об'єм мережі полегшує процес навчання.

Перевагами мереж із зворотними зв'язками є складність навчання, викликана великим числом нейронів для алгоритмів одного і того ж рівня складності. Недоліки цього виду мереж - потрібні спеціальні умови, що гарантують сходимость обчислень.

- Мережі *прямого розповсюдження і рекурентні мережі*.

До мереж прямого розповсюдження зазвичай відносять перцептрони, мережу Back Propagation, мережу зустрічного розповсюдження та карти Кохонена. Всі зв'язки направлені строго від вхідних нейронів до вихідних.

Характерна особливість рекурентних мереж - наявність блоків динамічної затримки і зворотних зв'язків, що дозволяє їм обробляти динамічні моделі, наприклад, мережа Хопфілда - мережа, що складається з двох шарів, в якій прихований шар охоплений динамічним зворотним зв'язком, що дозволяє врахувати передісторію спостережуваних процесів і накопичити інформацію для вироблення правильної стратегії управління. Ці мережі застосовуються в системах управління рухомими об'єктами. Окремим випадком рекурентних мереж є двонаправлені мережі. У таких мережах між шарами існують зв'язки як в напрямі від вхідного шару до вихідному, так і в зворотному.

Хоча створені мережі всіх конфігурацій, які тільки можна собі представити, пошарова організація нейронів копіює шаруваті структури певних відділів мозку. Виявилось, що такі багатошарові мережі володіють більшими можливостями ніж одношарові, і останніми роками були розроблені алгоритми для їх навчання. Багатошарові мережі можуть будуватися з каскадів шарів. Вихід одного шару є входом для подальшого шару.

На жаль, немає загальноприйнятого способу підрахунку числа шарів в мережі.

3. Навчання нейромережі. Зворотнє поширення похибки.

Перед використанням нейронної мережі її необхідно навчити. Процес навчання нейронній мережі полягає в підстроюванні її внутрішніх параметрів під конкретне завдання. Алгоритм роботи нейронної мережі є ітеративним, його кроки називають епохами або циклами.

Епоха - одна ітерація в процесі навчання, що включає пред'явлення всіх прикладів з навчальної множини і, можливо, перевірку якості навчання на контрольній множині.

Процес навчання здійснюється на навчальній вибірці. Вона включає вхідні значення і відповідні їм вихідні значення набору даних. В ході навчання нейронна мережа знаходить деякі залежності вихідних полів від вхідних. Таким чином, перед нами ставиться питання - які вхідні поля (ознаки) нам необхідно використовувати. Спочатку вибір здійснюється евристично, далі кількість входів може бути змінена.

Аналітик повинен визначити кількість шарів в мережі і кількість нейронів в кожному шарі. Далі необхідно призначити такі значення вагів і зміщень, які зможуть мінімізувати помилку рішення. Ваги і зміщення автоматично підстроюються так, щоб мінімізувати різницю між бажаними і отриманими на виході сигналами, яка називається помилка навчання.

Помилка навчання для побудованої нейронної мережі обчислюється шляхом порівняння вихідних і цільових (бажаних) значень. З отриманих різниць формується функція помилок. *Функція помилок* - це цільова функція, що вимагає мінімізації в процесі керованого навчання нейронної мережі. За допомогою функції помилок можна оцінити якість роботи нейронної мережі під час навчання. Наприклад, часто використовується сума квадратів помилок. Від якості навчання нейронній мережі залежить її здатність вирішувати поставлені перед нею завдання.

Навчання з вчителем передбачає, що для кожного навчального вхідного прикладу потрібне знання правильної відповіді або функції оцінки якості відповіді. Таке навчання називають *керованим*. Нейронній мережі пред'являються значення вхідних і вихідних сигналів, а вона по певному алгоритму підстроює ваги синаптичних зв'язків. В процесі навчання проводиться корегування вагів мережі за наслідками порівняння фактичних вихідних значень з вхідними, відомими наперед.

При *навчанні без вчителя* розкривається внутрішня структура даних або кореляції між зразками в наборі даних. Виходи нейронної мережі формуються самостійно, а ваги змінюються по алгоритму, що враховує тільки вхідні і похідні від них сигнали. Це навчання називають також *некерованим*. В результаті такого навчання об'єкти або приклади розподіляються по категоріях, самі категорії і їх

кількість можуть бути наперед не відомі.

При навчанні нейронних мереж часто виникають серйозні труднощі, названі *проблемою перенавчання* (overfitting). Перенавчання, або надмірно близька підгонка - надмірно точна відповідність нейронної мережі конкретному набору навчальних прикладів, при якому мережа втрачає здібність до узагальнення.

Перенавчання виникає в разі дуже довгого навчання, недостатнього числа навчальних прикладів або переускладненої структури нейронної мережі. Перенавчання пов'язане з тим, що вибір навчальної (тренувальної) множини є випадковим. З перших кроків навчання відбувається зменшення помилки. На подальших кроках з метою зменшення помилки (цільовій функції) параметри підстроюються під особливості навчальної множини. Проте при цьому відбувається «підстроювання» не під загальні закономірності ряду, а під особливості його частини - навчальної підмножини. При цьому точність прогнозу зменшується. Один з варіантів боротьби з перенавчанням мережі - ділення навчальної вибірки на дві множини (навчальної і тестової). На навчальній множині відбувається навчання нейронній мережі. На тестовій множині здійснюється перевірка побудованої моделі. Ці множини не повинні перетинатися. З кожним кроком параметри моделі змінюються, проте постійне зменшення значення цільової функції відбувається саме на навчальній множині. При розбитті множини на дві ми можемо спостерігати зміну помилки прогнозу на тестовій множині паралельно із спостереженнями над навчальною множиною. Якась кількість кроків помилки прогнозу зменшується на обох множинах. Проте на певному кроці помилка на тестовій множині починає зростати, при цьому помилка на навчальній множині продовжує зменшуватися.

Цей момент вважається кінцем реального або справжнього навчання, з нього і починається перенавчання. Прогноз на тестовій множині є перевіркою працездатності побудованої моделі. Помилка на тестовій множині може бути помилкою прогнозу, якщо тестова множина максимально наближена до теперішнього моменту.

В процесі навчання НМ одним із найпоширеніших методів навчання є метод зворотного поширення помилки.

Метод зворотного розповсюдження помилки. Для навчання багатошарових мереж був запропонований градієнтний алгоритм навчання з вчителем, який проводить сигнал помилки, обчислений виходами персептрона, до його входів, шар за шаром. Зараз це найпопулярніший метод навчання багатошарових персептронів. Його перевага в тому, що він може навчити всі шари нейронної мережі, і його легко прорахувати локально. Проте цей метод є дуже довготривалим, до того ж, для його застосування потрібно, щоб передавальна

функція нейронів була такою, що диференціюється. При цьому в персептронах довелося відмовитися від бінарного сигналу, і користуватися на вході безперервними значеннями.

Алгоритм навчання мережі на основі метода зворотного розповсюдження помилки вимагає виконання наступних операцій:

1. Вибрати чергову навчальну пару з навчальної множини та подати вхідний вектор на вхід мережі.
2. Обчислити вихід мережі.
3. Обчислити різницю між виходом мережі і необхідним виходом (цільовим вектором навчальної пари).
4. Підкоригувати ваги мережі так, щоб мінімізувати помилку.

Повторювати кроки з 1 по 4 для кожного вектора навчальної множини до тих пір, поки помилка на всій множині не досягне прийняттого рівня.

Зворотне поширення помилки. Зміна величини кроку. Локальні мінімуми.

В процесі навчання НМ методом зворотного поширення помилки є декілька «наріжних» каменів, на які треба звернути увагу.

Привабливою рисою нейрокомп'ютінга є єдиний принцип навчання нейромереж - *мінімізація функції помилки*. Мережа починає поступово модифікувати свою конфігурацію - стан всіх своїх синаптичних ваг - так, щоб мінімізувати цю помилку. У результаті, в процесі навчання мережа все краще справляється з покладеним на неї завданням.

Не вдаючись до математичних тонкощів, образно цей процес можна уявити собі як пошук мінімуму функції помилки $E(w)$, залежної від набору всіх синаптичних ваг мережі w (рис. 9.6).

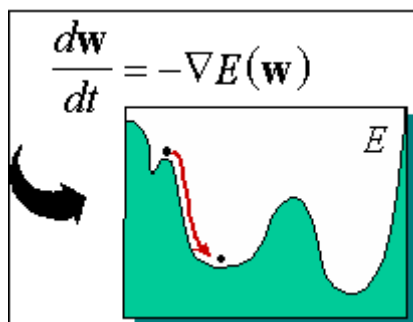


Рис. 9.6. Навчання мережі як задача оптимізації.

Базовою ідеєю всіх алгоритмів навчання є облік локального градієнта в просторі конфігурацій для вибору траєкторії якнайшвидшого спуску по функції помилки. Функція помилки, проте, може мати безліч локальних мінімумів, що представляють суб-оптимальні рішення. Ідеальний метод навчання повинен

знайти глобальний оптимум конфігурації мережі.

Першим критичним параметром є величина кроку, на яку ми змінюємо ваги. Якщо б ми корегували кожен раз силу зв'язків між нейронами так, щоб мінімізувати помилку до 0, наша НМ мало б чого навчилася, оскільки б вона «зависла» в середньостатистичному локальному мінімумі і не могла б знайти оптимальну точку, в якій максимально точно прогнозує вихідні дані.

На певній гіперповерхні можуть бути заглибини, в яких з усіх сторін точки знаходитимуться вище, тобто всюди навколо помилка є вищою. Але дана точка не є найглибшою точкою на поверхні помилки. Відповідно, якщо система не зможе робити певний крок більшим, ніж певне відхилення, вона не зможе вийти з локального мінімуму.

Коли ми маємо справу із функцією із мільйоном параметрів, а НМ є саме такими функціями, то поверхня нашої функції помилки є досить неоднорідною і має велику кількість заглиблень, жолобів, і «застрягти» в певному локальному мінімумі – досить легко. Якщо крок зміни системи буде досить маленький, то ми не можемо зробити достатній ривок, щоб вистрибнути з мінімуму. Якщо крок буде занадто великим, система стрибатиме постійно між досить віддаленими точками на поверхні помилки і ймовірність того, що вона потрапить в глобальний чи достатній локальний мінімум – дуже низька.

Які є методи оптимізації процесу знаходження локальних мінімумів?

Адаптивна зміна величини кроку.

Ми спостерігаємо, на скільки в процесі навчання зменшилася наша помилка. Якщо крок за кроком, збільшуючи величину кроку, наша НМ перестає зменшувати помилку, це означає, що далі крок необхідно зменшувати. Ми починаємо зменшувати розміру кроку, тримаючи темп зменшення помилки, щоб він коливався в певному невеликому діапазоні і не «стрибав» вгору або вниз. Таким чином ми поступово зменшуємо і збільшуємо крок, з яким рухаємось, і, таким чином, збільшуємо вірогідність встановлення навіть глобального мінімуму.

Підсумовуючи: ***Метод зворотного поширення помилки*** – спосіб корегування вагів в НМ таким чином, щоб внаслідок корегування наша НМ видавала меншу помилку на даному прикладі. Ітерація за ітерацією, надаючи нейронній мережі приклад за прикладом і корегуючи відповідним чином ваги в НМ пропорційно до величини помилки, ми наближаємо вектор зв'язків в НМ до такого вигляду, в якому НМ буде видавати максимально репрезентативні дані, які відповідатимуть нашим очікуванням.

Одними із ключових параметрів, які впливають на процес навчання – це величина кроку, методи зміни кроку, а також метод ініціалізації початкових значень вагів в НМ.