Аналіз нейронних алгоритмів

НАДРИГАЙЛО Т.Ж., МОЛЧАНОВА К.А.

Дніпродзержинський державній технічний університет

У даної роботі розглядаються нейронні мережі Хопфілда та Хемінга, нейронна мережа Хеба та алгоритм оберненого поширення, їх застосування та методи їх використання. Представлено опис програми розпізнавання рукописних символів із зображення. Для розв'язання задачі використано алгоритм оберненого поширення, який реалізований на мові Visual Studio.NET C#.

В данной работе рассматриваются нейронные сети Хопфилда и Хемминга, нейронная сеть Хеба и алгоритм обратного распространения, их применение и методы их использования. Представлено описание программы распознавания рукописных символов с изображения. Для решения задачи использован алгоритм обратного распространения, который реализован на языке Visual Studio.NET C#.

In this paper are considered neural network Hopfield and Hamming, Heba's neural network and back-propagation algorithm and their application and methods of their use. It was represented of software for the description of recognition handwriting symbols from the image. To solve the problem was used the back-propagation algorithm, which was implemented in language Visual Studio.NET C#.

Термін "нейронні мережі" сформувався до середини 50-х років XX століття. Основні результати в цій області пов'язані з іменами У.Маккалоха, Д.Хеба, Ф.Розенблатта, М.Мінського, Дж.Хопфілда.

Під нейронними мережами розуміють обчислювальні структури, що моделюють прості біологічні процеси, зазвичай асоційовані з процесами людського мозку. Вони являють собою розподілені і паралельні системи, здатні до адаптивного навчання шляхом аналізу позитивних і негативних впливів.

Штучні нейронні мережі будуються по принципах організації і функціонування їхніх біологічних аналогів. Глибоке вивчення штучних нейронних мереж вимагає знання нейрофізіології, науки про пізнання, психології, фізики (статистичної механіки), теорії керування, теорії обчислень, проблем штучного інтелекту, математики, розпізнавання образів, рівнобіжних обчислень і апаратних засобів (цифрових і аналогових). З іншого боку, штучні нейронні мережі також стимулюють ці дисципліни, забезпечуючи їх новими інструментами і уявленнями. Цей симбіоз життєво необхідний для дослідження нейронних мереж. Вони здатні вирішувати широке коло задач розпізнавання образів, ідентифікації, прогнозування, оптимізації, керування складними об'єктами.

У даній роботі розглянуто задачу по розпізнаванню символів, заданих за допомогою зображення.

Постановка задачі. Необхідно дослідити та проаналізувати алгоритми роботи та побудови нейронних мереж. Розглянути нейрону мережу Хопфілда та Хемінга, алгоритм оберненого поширення та нейрону мережу Хеба. На основі розглянутих мереж, використовуючи один з трьох алгоритмів, розробити програму розпізнавання рукописних символів.

Реалізація алгоритму розпізнавання тексту досить складна і складається з цілого ряду взаємопов'язаних блоків, серед яких блоки препроцесування, сегментації, виділення характеристик, класифікації та контекстуальної обробки. Паперовий документ сканується і створюється зображення у відтінках сірого кольору або бінарне (чорно-біле) зображення. На стадії препроцесування застосовується фільтрація для видалення шуму, область тексту локалізується і перетвориться до бінарного зображення за допомогою глобального і локального адаптивного порогового перетворювача. На кроці сегментації зображення тексту розділяється на окремі символи. Це завдання особливо важке для рукописного тексту, яке містить зв'язки між сусідніми символами. Один з ефективних прийомів полягає в розчленуванні складеного зразка на малі зразки (проміжна сегментація) і знаходженні точок правильної сегментації з використанням виходу класифікатора за зразками. Внаслідок різного нахилу, перекручень, перешкод і стилів листа розпізнавання сегментованих символів є непростим завданням.

Вирішення цієї задачі за допомогою нейронних мереж розподілена так: відбувається неявне витяг характеристик всередині самої нейромережі, в цьому випадку виділення ознак та їх класифікація об'єднані, проходячи поряд з навчанням мережі. Таким чином, мережа може звикнути до стилю тексту, що дає можливість отримати оптимальні результати. Але в будь-якому випадку, здатності самої людини до розпізнавання тексту (повна незалежність від вищезгаданих чинників) не можуть йти ні в яке порівняння навіть з найсучаснішими OCR-системами.

Однак не слід упускати з виду і те, що НМ не є панацеєю від проблем такого типу, вони також можуть помилятися. НМ лише можуть допомогти при вирішенні важко формалізованих задач, вимагаючи висококласного фахівця з розробки архітектури мережі, здатного вирішувати питання, пов'язані з проектуванням, використанням спеціалізованого програмного забезпечення і т.д.

Нейронні мережі Хопфілда. Серед різних конфігурацій штучних НМ зустрічаються такі, при класифікації яких за принципом навчання, строго кажучи, не підходять ні навчання з учителем, ні навчання без учителя. У таких мережах вагові коефіцієнти синапсів розраховуються тільки один раз перед початком функціонування мережі на основі інформації оброблюваних даних, і все навчання мережі зводиться саме до цього розрахунку. З одного боку, пред'явлення апріорної інформації можна розцінювати, як допомога вчителя, але з іншого – мережа фактично просто запам'ятовує зразки до того, як на її вхід надходять реальні дані, і не може змінювати свою поведінку, тому говорити про ланку зворотного зв'язку з "світом" (вчителем) не доводиться. З мереж з подібною логікою роботи найбільш відомі

© Надригайло Т.Ж., Молчанова К.А., 2011

мережа Хопфілда та мережа Хемінга, які зазвичай використовуються для організації асоціативної пам'яті.

Структурна схема мережі Хопфілда наведена на рис. 1. Вона складається з єдиного шару нейронів, число яких є одночасно числом входів і виходів мережі. Кожен нейрон зв'язаний синапсами з усіма іншими нейронами, а також має один вхідний синапс, через який здійснюється введення сигналу. Вихідні сигнали, як зазвичай, утворюються на аксонах.

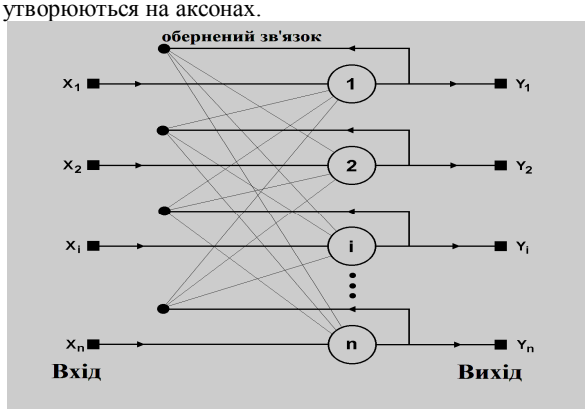

Рис. 1. Структурна схема мережі Хопфілда

Задача, яка вирішується цією мережею в якості асоціативної пам'яті, як правило, формулюється наступним чином. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують деякі об'єкти або характеристики процесів), які вважаються зразковими. Мережа повинна вміти з довільного неідеального сигналу, поданого на її вхід, виділити ("згадати" по частковій інформації) відповідний зразок (якщо такий є) або "дати висновок" про те, що вхідні дані не відповідають жодному зі зразків. У загальному випадку, будь-який сигнал може бути описаний вектором $X = \{x_i : i = 0...n-1\}$, n – число нейронів у мережі і розмірність вхідних та вихідних векторів. Кожен елемент x_i дорівнює або 1, або -1. Позначимо вектор, що описує k-ий зразок, через X^k, а його компоненти, відповідно, $-x_i^k$, $k = 0,...,m-1$, m – число зразків. Коли мережа розпізнає (або "згадає") будь-який зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = X^k$, де Y – вектор вихідних значень мережі: $Y = \{y_i : i = 0,...,n-1\}$. В іншому випадку, вихідний вектор не співпаде з жодним зразковим.

Якщо, наприклад, сигнали являють собою якісь зображення, то, відобразивши в графічному вигляді дані з виходу мережі, можна буде побачити картинку, повністю співпадаючу з одним зі зразкових (у разі успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

На стадії ініціалізації мережі вагові коефіцієнти синапсів встановлюються наступним чином:

$$w_{ij} = \begin{cases} \sum_{k=0}^{m-1} x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases}. \qquad (1)$$

Тут i і j – індекси, відповідно, предсинаптичного і постсинаптичного нейронів; x_i^k, x_j^k – i-ий і j-ий елементи вектора k-ого зразка.

Алгоритм функціонування мережі наступний (p - номер ітерації):

1. На входи мережі подається невідомий сигнал. Фактично його введення здійснюється безпосередньо установкою значень аксонів:

$$y_i(0) = x_i, \quad i = 0...n-1, \qquad (2)$$

тому позначення на схемі мережі вхідних синапсів в явному вигляді носить чисто умовний характер. Нуль в дужках справа від y_i означає нульову ітерацію в циклі роботи мережі.

2. Розраховується новий стан нейронів

$$s_j(p+1) = \sum_{i=0}^{n-1} w_{ij} y_i(p), \quad j = 0..n-1 \qquad (3)$$

і нові значення аксонів

$$y_j(p+1) = f[s_j(p+1)], \qquad (4)$$

де f – активаційна функція у вигляді стрибка, наведена на рис. 2а.

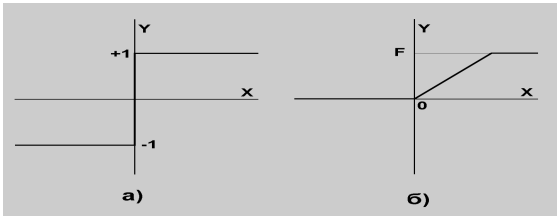

Рис. 2. Активаційні функції

3. Перевірка, чи змінилися вихідні значення аксонів за останню ітерацію. Якщо так – перехід до пункту 2, інакше (якщо виходи застабілізувалися) – кінець. При цьому вихідний вектор являє собою зразок, найкращим чином поєднується з вхідними даними.

Нейронні мережі Хемінга. Як говорилося вище, іноді мережа не може провести розпізнавання і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей мережі. Для мережі Хопфілда число запам'ятовуючих образів m не повинно перевищувати величини, приблизно рівної $0.15 \cdot n$. Крім того, якщо два образи А і Б сильно схожі, вони, можливо, будуть викликати у мережі перехресні асоціації, тобто пред'явлення на входи мережі вектора А призведе до появи на її виходах вектора Б і навпаки.

Коли немає необхідності, щоб мережа в явному вигляді видавала зразок, тобто досить, скажімо, отримувати номер зразка, асоціативну пам'ять успішно реалізує мережа Хемінга. Дана мережа характеризується, в порівнянні з мережею Хопфілда, меншими витратами на пам'ять і обсягом обчислень, що стає очевидним з її структури (рис. 3).

Мережа складається з двох шарів. Перший і другий шари мають по m нейронів, де m – число зразків. Нейрони першого шару мають по n синапсів, з'єднаних із входами мережі (утворюючими фіктивний нульовий шар). Нейрони другого шару пов'язані між собою інгібіторними (негативними зворотними) синаптичними зв'язками. Єдиний синапс з позитивним зворотним зв'язком для кожного нейрона з'єднаний з його ж аксоном.

Ідея роботи мережі полягає в знаходженні відстані Хемінга від тестованого образу до всіх зразків. Відстанню Хемінга називається число відрізняючихся бітів в двох бінарних векторах. Мережа має вибрати зразок з мінімальною відстанню Хемінга до невідомого вхідного

сигналу, в результаті чого буде активізований тільки один вихід мережі, відповідний цим зразком.

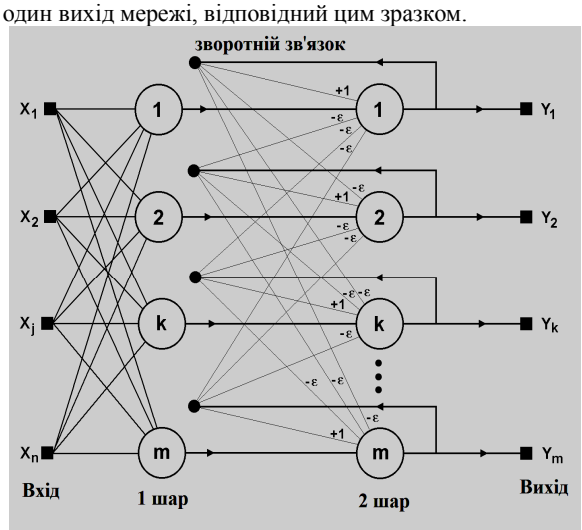

Рис. 3. Структурна схема мережі Хемінга

На стадії ініціалізації ваговим коефіцієнтам першого шару і порогу активаційній функції присвоюються наступні значення:

$$w_{ik} = \frac{x_i^k}{2}, \quad i = 0...n-1, \quad k = 0...m-1, \qquad (5)$$

$$T_k = \frac{n}{2}, \quad k = 0...m-1. \qquad (6)$$

Тут x_i^k – i-ий елемент k-ого зразка.

Вагові коефіцієнти гальмуючих синапсів у другому шарі беруть рівними деякій величині $0 < \varepsilon < 1/m$. Синапс нейрона, пов'язаний з його ж аксоном має вагу +1.

Алгоритм функціонування мережі Хемінга наступний:

1. На входи мережі подається невідомий вектор $X = \{x_i : i = 0...n-1\}$, виходячи з якого розрахо-вуються стани нейронів першого шару (верхній індекс у дужках вказує номер шару):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j = 0...m-1. \qquad (7)$$

Після цього отриманими значеннями ініціалізуються значення аксонів другого шару:

$$y_j^{(2)} = y_j^{(1)}, \quad j = 0...m-1. \qquad (8)$$

2. Обчислити нові стани нейронів другого шару:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), \; k \neq j, \; j = 0...m-1 \qquad (9)$$

і значення їх аксонів:

$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], \quad j = 0...m-1. \qquad (10)$$

Активаційна функція f має вигляд порогу (рис. 2б), причому величина F повинна бути досить великою, щоб будь-які можливі значення аргументу не призводили до насичення.

3. Перевірити, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так – перейди до кроку 2. Інакше – кінець.

З оцінки алгоритму видно, що роль першого шару досить умовна: скориставшись один раз на кроці 1 значеннями його вагових коефіцієнтів, мережа більше не звертається до нього, тому перший шар може бути взагалі виключений з мережі (замінений на матрицю вагових коефіцієнтів).

Обговорення мереж, що реалізують асоціативну пам'ять, було б неповним без хоча б короткої згадки про двобічну асоціативну пам'ять (ДАП). Вона є логічним розвитком парадигми мережі Хопфілда, до якої для цього достатньо додати другий шар.

Мережа здатна запам'ятовувати пари асоційованих один з одним образів. Нехай пари образів записуються у вигляді векторів $X^k = \{x_i^k : i = 0...n-1\}$ і $Y^k = \{y_j^k : j = 0...m-1\}, k = 0...r-1$, де r – число пар. Подача на вхід першого шару деякого вектора $P = \{p_i : i = 0...n-1\}$ викликає утворення на вході другого шару якогось іншого вектора $Q = \{q_j : j = 0...m-1\}$, який потім знову надходить на вхід першого шару. При кожному такому циклі вектора на виходах обох шарів наближаються до пари зразкових векторів, перший з яких – X – найбільш схожий на P, який був поданий на вхід мережі на самому початку, а другий – Y – асоційований з ним. Асоціацію між векторами кодуються у ваговій матриці $W^{(1)}$ першого шару. Вагова матриця другого шару дорівнює транспонованої першої $\left(W^{(1)}\right)^{(T)}$. Процес навчання, також як і у випадку мережі Хопфілда, полягає в попередньому розрахунку елементів матриці W за формулою:

$$w_{ij} = \sum_k x_i y_j, i = 0...n-1, j = 0...m-1. \qquad (11)$$

Алгоритм оберненого поширення. Серед різних структур нейронних мереж (НМ) однієї з найбільш відомих є багатошарова структура, в якій кожен нейрон довільного шару пов'язаний з усіма аксонами нейронів попереднього шару або, у разі першого шару, з усіма входами НМ. Такі НМ називаються повнозв'язні. Коли в мережі тільки один шар, алгоритм її навчання з учителем досить очевидний, оскільки правильні вихідні стани нейронів єдиного шару завідомо відомі, і підстроювання синаптичних зв'язків йде в напрямку, що мінімізує помилку на виході мережі. За цим принципом будується, наприклад, алгоритм навчання одношарового перцептрона. У багатошарових же мережах оптимальні вихідні значення нейронів всіх верств, крім останнього, як правило, не відомі, і двох чи більше шаровий перцептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах НМ. Один з варіантів вирішення цієї проблеми – розробка наборів вихідних сигналів, що відповідають вхідним, для кожного шару НМ, що, звичайно, є дуже трудомісткою операцією і не завжди здійснено. Другий варіант – динамічне підстроювання вагових коефіцієнтів синапсів, в ході якої вибираються, як правило, найбільш слабкі зв'язки і змінюються на малу величину в ту чи іншу сторону, а зберігаються тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Очевидно, що даний метод "тику", незважаючи на свою простоту, вимагає великих рутинних обчислень. І, нарешті, третій, більш близький до істини варіант – поширення сигналів помилки від виходів НМ до її входів, у напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Цей

алгоритм навчання НМ отримав назву процедури зворотного поширення. Саме він буде розглянутий надалі.

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

$$E(w)=\frac{1}{2}\sum_{j,p}(y_{j,p}^{(N)}-d_{j,p})^2\ , \qquad (12)$$

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 вихідний стан цього нейрона.

Підсумовування ведеться за всіма нейронам вихідного шару і по всім оброблюваним мережею образам. Мінімізація ведеться методом градієнтного спуску, що означає підстроювання вагових коефіцієнтів наступним чином:

$$\Delta w_{ij}^{(n)}=-\eta\cdot\frac{\partial E}{\partial w_{ij}}\ . \qquad (13)$$

Тут w_{ij} – ваговий коефіцієнт синаптичного зв'язку, що з'єднує i-ий нейрон шару $(n–1)$ з j-им нейроном шару n, η – коефіцієнт швидкості навчання, $0<\eta<1$.

Як показано у (13):

$$\frac{\partial E}{\partial w_{ij}}=\frac{\partial E}{\partial y_j}\cdot\frac{dy_j}{ds_j}\cdot\frac{\partial s_j}{\partial w_{ij}}\ . \qquad (14)$$

Тут під y_j, як і раніше, мається на увазі вихід нейрона j, а під s_j – зважена сума його вхідних сигналів, тобто аргумент активаційної функції. Так як множник dy_j/ds_j є похідною цієї функції по її аргументу, з цього випливає, що похідна активаційної функції повинна бути визначена на всій вісі абсцис. У зв'язку з цим функція одиничного стрибка і інші активаційні функції з неоднорідностями не підходять для розглянутих НМ. У них застосовуються такі гладкі функції, як гіперболічний тангенс або класичний сигмоїд з експонентою. У випадку гіперболічного тангенса

$$\frac{dy}{ds}=1-s^2\ . \qquad (15)$$

Третій множник $\partial s_j/\partial w_{ij}$, очевидно, дорівнює виходу нейрона попереднього шару $y_i^{(n-1)}$.

Що стосується першого множника в (14), він легко розкладається наступним чином:

$$\frac{\partial E}{\partial y_j}=\sum_k\frac{\partial E}{\partial y_k}\cdot\frac{dy_k}{ds_k}\cdot\frac{\partial s_k}{\partial y_j}=\sum_k\frac{\partial E}{\partial y_k}\cdot\frac{dy_k}{ds_k}\cdot w_{jk}^{(n+1)}\ . \qquad (16)$$

Тут підсумовування по k виконується серед нейронів шару $n+1$.

Вводимо нову змінну

$$\delta_j^{(n)}=\frac{\partial E}{\partial y_j}\cdot\frac{dy_j}{ds_j}\ . \qquad (17)$$

Ми отримаємо рекурсивну формулу для розрахунків величин $\delta_j^{(n)}$ шару n з величин $\delta_k^{(n+1)}$ більш старшого шару $n+1$.

$$\delta_j^{(n)}=\left[\sum_k\delta_k^{(n+1)}\cdot w_{jk}^{(n+1)}\right]\cdot\frac{dy_j}{ds_j}\ . \qquad (18)$$

Для вихідного слою:

$$\delta_l^{(N)}=(y_l^{(N)}-d_l)\cdot\frac{dy_l}{ds_l}\ . \qquad (19)$$

Тепер ми можемо записати (13) в розкритому вигляді:

$$\Delta w_{ij}^{(n)}=-\eta\cdot\delta_j^{(n)}\cdot y_i^{(n-1)}\ . \qquad (20)$$

Іноді для надання процесу корекції ваг деякої інерційності, що згладжує різкі скачки при переміщенні по поверхні цільової функції, (20) доповнюється значенням зміни ваги на попередній ітерації

$$\Delta w_{ij}^{(n)}(t)=-\eta\cdot(\mu\cdot\Delta w_{ij}^{(n)}(t-1)+(1-\mu)\cdot\delta_j^{(n)}\cdot y_i^{(n-1)}) \qquad (21)$$

де μ – коефіцієнт інерційності, t – номер поточної ітерації.

Таким чином, повний алгоритм навчання НМ за допомогою процедури зворотного поширення будується так:

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

$$s_j^{(n)}=\sum_{i=0}^{M}y_i^{(n-1)}\cdot w_{ij}^{(n)}\ , \qquad (22)$$

де M – число нейронів у шарі $n-1$ з урахуванням нейрона з постійним вихідним станом $+1$, що задає зсув;

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

$y_j^{(n)}=f(s_j^{(n)})$,

де $f()$ – сигмоїд;

$y_q^{(0)}=I_q$,

де I_q – q-а компонента вектора вхідного образу.

2. Розрахувати $\delta^{(N)}$ для вихідного шару за формулою (2.19). Розрахувати за формулою (20) або (21) зміни ваг $\Delta w^{(N)}$ шару N.

3. Розрахувати за формулами (18) і (20) (або (18) і (21)) відповідно $\delta^{(n)}$ и $\Delta w^{(n)}$ для всіх інших шарів, $n=N-1, \ldots 1$.

4. Скорегувати всі ваги в НМ

$$w_{ij}^{(n)}(t)=w_{ij}^{(n)}(t-1)+\Delta w_{ij}^{(n)}(t)\ . \qquad (23)$$

5. Якщо помилка мережі істотна, перейти на крок 1. В іншому випадку – кінець.

НМ на кроці 1 поперемінно у випадковому порядку пред'являються всі тренувальні образи, щоб мережа, образно кажучи, не забувала одні в міру запам'ятовування інших. Алгоритм ілюструється рисунком 4.

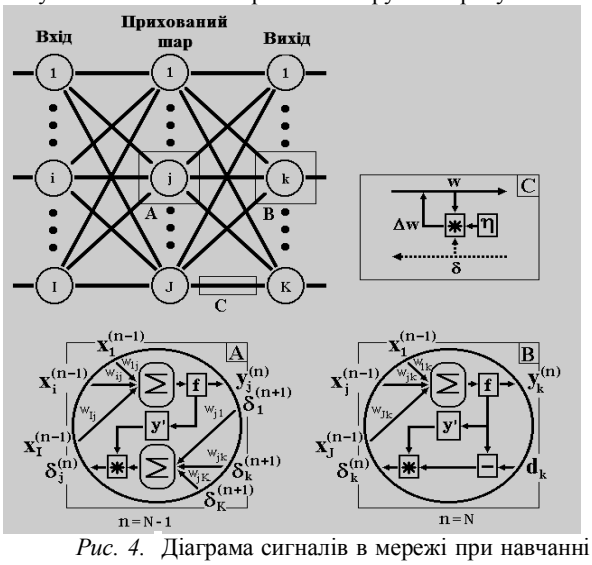

Рис. 4. Діаграма сигналів в мережі при навчанні за алгоритмом зворотного поширення

З виразу (20) випливає, що коли вихідне значення $y_i^{(n-1)}$ прагне до нуля, ефективність навчання помітно знижується. При подвійних вхідних векторах в середньому половина вагових коефіцієнтів не буде корегуватися, тому область можливих значень виходів нейронів

[0,1] бажано зсунути в межі [-0.5, +0.5], що досягається простими модифікаціями логістичних функцій. Наприклад, сигмоїд з експонентою перетвориться до виду:

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha \cdot x}}. \qquad (24)$$

Тепер розглянемо питання ємності НМ, тобто числа образів, які подавали на її входи, які вона здатна навчитися розпізнавати. Для мереж з числом шарів більше двох, він залишається відкритим. Для НМ з двома шарами, тобто вихідним і одним прихованим шаром, детерміністська ємність мережі C_d оцінюється так:

$$N_w/N_y < C_d < N_w/N_y \cdot log(N_w/N_y), \qquad (25)$$

де N_w – число ваг, які підбираються, N_y – число нейронів в вихідному шарі.

Слід зазначити, що даний вираз отримано з урахуванням деяких обмежень. По-перше, число входів N_x і нейронів в прихованому шарі N_h має задовольняти нерівності $N_x+N_h>N_y$. По-друге, $Nw/Ny>1000$. Проте вищенаведена оцінка виконувалася для мереж з активаційними функціями нейронів у вигляді порогу, а ємність мереж з гладкими активаційними функціями, наприклад (24), зазвичай більше. Крім того, в назві об'єм фігурує прикметник "детерміністський" означає, що отримана оцінка об'єму підходить абсолютно для всіх можливих вхідних образів, які можуть бути представлені N_x входами. Насправді розподіл вхідних образів, як правило, має певну регулярність, що дозволяє НМ проводити узагальнення і, таким чином, збільшувати реальну ємність. Так як розподіл образів, в загальному випадку, заздалегідь не відомо, ми можемо говорити про таку ємність тільки ймовірно, але зазвичай вона рази в два перевищує ємність детерміністську.

Розглянута НМ має кілька "вузьких місць". По-перше, в процесі навчання може виникнути ситуація, коли великі позитивні або негативні значення вагових коефіцієнтів змістять робочу точку на сигмоїді багатьох нейронів в область насичення. По-друге, застосування методу градієнтного спуску не гарантує, що буде знайдено глобальний, а не локальний мінімум цільової функції. Ця проблема пов'язана ще з однією, а саме – з вибором величини швидкості навчання [2].

Нейрона мережа Хеба. Головна риса, що робить навчання без вчителя привабливим, – це його "самостійність". Процес навчання, як і у разі навчання з учителем, полягає в підстроюванні ваг синапсів. Деякі алгоритми, правда, змінюють і структуру мережі, тобто кількість нейронів і їх взаємозв'язок, але такі перетворення правильніше назвати більш широким терміном – самоорганізацією. Очевидно, що підстроювання синапсів може проводитися тільки на підставі інформації, доступної в нейроні, тобто його стану і вже наявних вагових коефіцієнтів. Виходячи з цього міркування і, що більш важливо, за аналогією з відомими принципами самоорганізації нервових клітин, побудовані алгоритми навчання Хеба.

Сигнальний метод навчання Хеба полягає у зміні ваг за наступним правилом:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)}, \qquad (26)$$

де $y_i^{(n-1)}$ – вихідне значення нейрона і шару $n-1$, $y_j^{(n)}$ – вихідне значення нейрона j шару n; $w_{ij}(t)$ і $w_{ij}(t-1)$ – ваговий коефіцієнт синапсу, що з'єднує ці нейрони, на ітераціях t і $t-1$ відповідно; α – коефіцієнт швидкості навчання. Тут і далі, для спільності, під n мається на увазі довільний шар мережі. При навчанні за цим методом посилюються зв'язки між збудженими нейронами.

Існує також і диференціальний метод навчання Хеба.

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot \left[y_i^{(n-1)}(t) - y_i^{(n-1)}(t-1)\right] \times \left[y_j^{(n)}(t) - y_j^{(n)}(t-1)\right]. \qquad (27)$$

Тут $y_i^{(n-1)}(t)$ і $y_i^{(n-1)}(t-1)$ – вихідне значення нейрона і шару $n-1$ відповідно на ітераціях t і $t-1$; $y_j^{(n)}(t)$ і $y_j^{(n)}(t-1)$ – те ж саме для нейрона j шару n. Як видно з формули (2.27), найсильніше навчаються синапси, що з'єднують нейрони ті, виходи яких найбільш динамічно змінилися в бік збільшення.

Повний алгоритм навчання із застосуванням вищенаведених формул буде виглядати так:

1. На стадії ініціалізації всім ваговим коефіцієнтам присвоюються невеликі випадкові значення.
2. На входи мережі подається вхідний образ, і сигнали збудження поширюються по всіх шарів згідно з принципами класичних прямопоточних (feedforward) мереж, тобто для кожного нейрона розраховується зважена сума його входів, до якої потім застосовується активаційна (передавальна) функція нейрона, в результаті чого виходить його вихідне значення $y_i^{(n)}$, $i = 0 \ldots M_i-1$, де M_i – число нейронів у шарі i; $n = 0 \ldots N-1$, а N – кількість шарів у мережі.
3. На підставі отриманих вихідних значень нейронів за формулою (26) або (27) робиться зміна вагових коефіцієнтів.
4. Цикл із кроку 2, поки вихідні значення мережі не стабілізуються із заданою точністю. Застосування цього нового способу визначення завершення навчання, відмінного від використання для мережі зворотного поширення, обумовлено тим, що підбір значення синапсів фактично не обмежений.

На другому кроці циклу поперемінно пред'являються всі образи з вхідного набору.

Слід зазначити, що вид відгуків на кожен клас вхідних образів не відомий заздалегідь і буде представляти собою довільне поєднання станів нейронів вихідного шару, обумовлене випадковим розподілом ваг на стадії ініціалізації. Разом з тим, мережа здатна узагальнювати схожі образи, відносячи їх до одного класу. Тестування навченої мережі дозволяє визначити топологію класів у вихідному шарі. Для приведення відгуків навченої мережі до зручного поданням можна доповнити мережу одним шаром, який, наприклад, за алгоритмом навчання одношарового перцептрона необхідно змусити відображати вихідні реакції мережі в необхідні образи [2].

В даний час існує велика кількість програм-обробників, що дозволяють вирішувати завдання розпізнавання образів, символів, тексту. При вирішенні цього завдання доводиться стикатися з цілою низкою проблем: шумозаглушення, виділення символів, блоків та їх подальша обробка. Добре, якщо документ писаний якісно та досить розбірливим почерком, але найчастіше доводиться стикатися з цілою низкою проблем. Наприклад, вам необхідно розпізнати документ, який рукописно написаний, але між літерами не має чіткого розділення, або під дуже сильним нахилом, з цими завданнями програми справляються з великими труднощами.

В розробленої програмі вхідні образи представляють собою чорно-біле зображення з чітко розділеними

рукописними символами. Навчання мережі фактично зводиться до завантаження і запам'ятовування ідеальних зображень. Потім на її вхід по черзі подаються зашумлені образи (рис. 5), які вона повинна розрізнити.

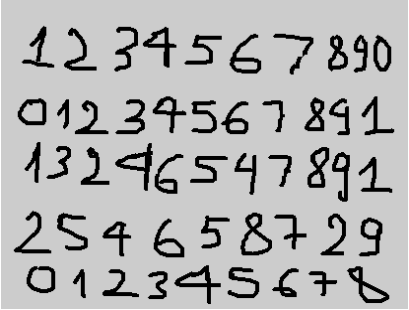

Рис. 5. Зразкові та тестові образи

Дослідивши три алгоритми нейронних мереж (нейронні мережі Хопфілда та Хемінга, алгоритм оберненого поширення та нейронні мережі Хеба), для реалізації поставленої задачі було вирішено використовувати алгоритм оберненого поширення.

Для навчання нейронної мережі ідеальним зображенням використовувалася база даних MNIST, яка містить 60000 зразків рукописних цифр. MNIST є укороченою версією бази NIST. Крім того зображення в базі нормовані за розміром і відцентровані усередині зображення [4].

Програма, що реалізує розпізнавання рукописних символів була розроблена у середовищі Visual Studio 2010 на мові C#.

Після запуску програми з'являється діалогове вікно, в якому необхідно вибрати у верхньому лівому кутку «Завантажити картинку» → «Параметри нейроної мережі» и у вспливаючиму діалоговому вікні вибрати та відкрити файл MNIST database.

Щоб завантажити зображення, яке необхідно розпізнати, необхідно повторити попередній крок, але вибрати не «Параметри нейронної мережі», а «Відкрити картинку» та вибрати необхідне зображення у вікні, яке з'явилося.

Після того, як буде вибране зображення почнеться процес розпізнавання. Програма розіб'є зображення на окремі символи та по одному буде їх розпізнавати. Цей процес можна спостерігати у блоці «Символ розпізнавання». Символи, які мережа вже обробила будуть по одному виводитись у полі «Результат».

Коли мережа закінчить процес розпізнавання, програма виділить, враховуючи відстань між словами та символи, які розташовуються не на одній лінії, усі на її думку словосполучення, на які було розбито зображення, у прямокутники з границею (рис. 6). Користувач буде мати змогу, за допомогою кнопок «Наступний» та «Попередній», передивитись усі символи, які були виділені та розпізнані.

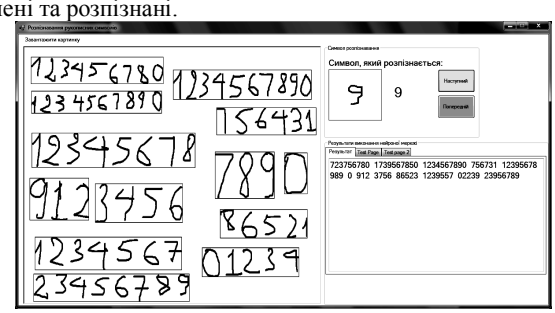

Рис. 6. Приклад розпізнавання зображення, де символи росташовані на різних рівнях

Таким чином, нейронні мережі, з використанням асоціативної пам'яті, дозволяють просто і ефективно розв'язати завдання відтворення образів по неповній та перекрученій інформації. Невисока ємність мереж (число запам'ятовуються образів) пояснюється тим, що, мережі не просто запам'ятовують образи, а дозволяють проводити їх узагальнення. Разом з тим, легкість побудови програмних і апаратних моделей роблять ці мережі привабливими для багатьох застосувань.

Висновки

У всіх розглянутих алгоритмах є свої позитивні сторони та як це не сумно і негативні. Наприклад, недоліком нейронних мереж Хопфілда та Хемінга є те, що вони базуються на основі асоціативної пам'яті, тобто з ростом кількості образів росте і використання пам'яті, а це досить великий мінус. До відомих недоліків алгоритму зворотного поширення відносяться: проблема параліча мережі, що може виникнути при її навчанні (це зв'язано з великими значеннями ваг), проблема локального мінімуму (мережа може потрапити в локальний мінімум, коли поруч є набагато більш глибокі мінімуми. статистичні методи навчання можуть допомогти уникнути цієї пастки, але вони дуже уповільнюють роботу розрахунків), розмір кроку (якщо розмір кроку фіксований і дуже малий, то збіжність надто повільна, якщо ж він фіксований і занадто великий, то може виникнути параліч або постійна нестійкість. Ефективно збільшувати крок до тих пір, поки не припиниться поліпшення оцінки і зменшувати, якщо такого покращення не відбувається).

Розроблена програма може бути корисна для розпізнавання номерів телефонів, різноманітних числових значень, а якщо мережу навчити ще і буквам, то можна розпізнавати любий рукописний текст, але зі збільшенням бази символів буде рости і час навчання мережі та розпізнавання нею символів. Також, дану програму можна модифікувати в сторону математичних виразів. Мається на увазі, щоб за допомогою нейронних мереж, на зображеннях, де представлені деякі математичні вирази (наприклад, визначений інтеграл або похідна), вони правильно розпізнавалися та одразу мали б можливість отримання їх результату.

ЛІТЕРАТУРА

1. Короткий С. Нейронные сети: алгоритм обратного распространения. – Режим доступу: http://masters.donntu.edu.ua/2009/fvti/trubarov/library/article2.htm
2. Короткий С. Нейронные сети: обучение без учителя. – Режим доступу: http://masters.donntu.edu.ua/2006/fvti/lazebnik/library/art13.htm
3. Інформація про базу даних MNIST. – Режим доступу: http://www.machinelearing.ru/wiki/index.php?title=MNIST_database_of_handwritten_digits
4. Знаходження та опис бази даних MNIST. – Режим доступу: http://yann.lecun.com/exdb/mnist/index.html
5. Круг П.Г. Нейронные сети и нейрокомпьютеры: Учебное пособие по курсу "Микропроцессоры". – М.: Издательство МЭИ, 2002. – 176 с.

пост. 25.05.11

© Надригайло Т.Ж., Молчанова К.А., 2011