

МЕТОДИ НАВЧАННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

© Щербина Ю.М., Годич О.В., 2001

Contains information about present situation of artificial neural networks learning concepts and methods. Advantages and disadvantages of these methods are analyzed. A new concept is purposed and described as an approach to avoiding known problems of learning with samples and of self-organized structures.

Описано сучасний стан у галузі навчання нейронних мереж. Пропонується нова концепція і описується підхід до проблем отримання знань з простими і самоорганізаційними структурами

Дослідження принципів побудови та функціонування штучних нейронних мереж є важливою та багатоаспектною проблемою, яка має велике теоретичне і прикладне значення. Фундаментальний огляд цих питань та перспективи розвитку проведено у працях [1, 2].

Важливим питанням практичної побудови штучної нейронної мережі (ШНМ) є її навчання, тобто адаптація до поставленої задачі.

ШНМ навчається згідно з правилами певного процесу, який модифікує її параметри (у більшості випадків це ваги семантичних зв'язків у ШНМ). Якщо мережа навчена, то, подавши на її вхід низку сигналів (вхідних векторів, образів), отримаємо відповідну низку правильних вихідних значень (реакцій) ШНМ.

Розглядають два класи навчальних методів: *детерміністський* та *стохастичний*.

Детерміністські методи покровоко здійснюють процедуру корекції ваг, яка базується на використанні їх поточних значень, значень входів, фактичних виходів і бажаних виходів.

Стохастичні методи навчання здійснюють псевдовипадкові зміни значень параметрів ШНМ, які спонукають до покращання значень на виході.

Для кожного із розглянутих класів розрізняють навчання з учителем та без учителя.

Навчання з учителем передбачає існування деякої множини пар вхід-вихід. "Учитель" подає вхідні образи, порівнює результуючі виходи із відповідними еталонними (бажаними), а потім корегує параметри ШНМ так, щоб зменшити різницю (відхилення) між виходом (реакцією) і відповідним еталоном.

Такий навчальний механізм аж ніяк не можна зіставити із існуючими аналогами у біологічних системах (наприклад, процес пізнання людиною довкілля). А отже, він відкидається дослідниками, які вважають, що ШНМ обов'язково повинні використовувати ті самі механізми, що й людський мозок.

При *навчанні без учителя* після подання вхідних образів ШНМ самоорганізується, корегуючи свої ваги за визначеним алгоритмом. Як наслідок відсутності вказаних еталонних виходів під час навчання результати непередбачливі з погляду визначення збуджувальних образів для конкретних нейронів. При цьому мережа організується (групується) у формі, яка відображає суттєві характеристики навчальної множини.

Для прикладних задач навчання з учителем виявилось досить важливим, незважаючи на його "неприродність". Тому почнемо розгляд методів навчання саме з них.

Нехай $(signal^t, target^t) \in Z$ — навчальна пара вхід-вихід, де Z — множина навчальних пар, $signal^t$ — вхідний образ, $target^t$ — еталон реакції ШНМ. Відзначимо, що $signal^t$ та $target^t$ є векторами (необов'язково однакової розмірності).

Навчання з учителем. При навчанні з учителем намагаються так перетворити параметри ШНМ, щоб відмінність між реакцією ШНМ та відповідним еталонним значенням з навчальної множини була якомога меншою. На практиці це зводиться до мінімізації так званої функції помилки від параметрів P ШНМ:

$$E(P) = \frac{1}{2} \sum_{i=1}^m (out_i^{last}(P) - t \arg et_i^t)^2.$$

Тут $t \arg et_i^t$ — i -та компонента еталонного вектора $t \arg et^t$, $out_i^{last}(P)$ — реакція i -го нейрона в останньому прошарку ШНМ, $P = (W, Q)$ — параметри, які є множиною W ваг семантичних зв'язків у ШНМ та множиною Q порогових рівнів реакції нейронів.

Для мінімізації функції $E(P)$ в принципі можна застосувати відомі методи. Наприклад, градієнтний метод, метод Ньютона тощо. На жаль, їх безпосереднє застосування до мінімізації $E(P)$ неможливе через надзвичайно велику розмірність задачі. Докладніше це буде розглянуто нижче.

Перший прийнятний спосіб мінімізації функції помилки запропонували Румельхарт, Гінт та Вільямс у 80-х роках [6]; він називається методом зворотного поширення помилки (back propagation of error). Цей метод є аналогом градієнтного методу, пристосованого для ШНМ. Його привабливість полягає у можливості паралельних корегувань параметрів ШНМ.

Метод зворотного поширення помилки та його модифікації. При навчанні на кожній ітерації будемо коректувати параметри у напрямку антиградієнта, тобто

$$\Delta P = -\varepsilon \nabla E(P). \quad (1)$$

Як бачимо, це є звичайний градієнтний метод, який гарантує збіжність до одного із локальних мінімумів.

Корекція параметрів відбувається на кожній ітерації, тобто на кожній ітерації потрібно обчислювати градієнт функції похибки та оптимальний крок. Постає питання обчислення градієнта при найменших обчислювальних затратах.

Найбільш очевидним, але не найкращим способом, є обчислення градієнта за означенням:

$$\frac{\partial E(P)}{\partial p_i} = \lim_{\Delta p_i \rightarrow 0} \frac{E(P + \Delta p_i) - E(P)}{\Delta p_i},$$

де Δp_i — приріст i -ї компоненти вектора параметрів P .

Але для розрахунку кожного значення функції $E(P)$ необхідно подати вхідний вектор і прорахувати вихідні значення усіх нейронів мережі, ще пов'язана з дуже великим обсягом обчислень. З огляду на те, що потрібно обчислювати всі компоненти градієнта таким способом, неефективність цього підходу стає очевидною.

У методі зворотного поширення помилки використано інший підхід — представлення $E(P)$ у вигляді складної функції і послідовне обчислення часткових похідних за формулами для складної функції.

Запишемо (1) для вагових коефіцієнтів:

$$\Delta w_{ij}^k = -\varepsilon \left(\frac{\partial E(W, Q)}{\partial w_{ij}^k} \right)_{W, Q},$$

де похідна обчислюється для поточних значень параметрів W, Q .

Для обчислення нових значень ваг застосовуємо формулу:

$$w_{ij}^k = w_{ij}^k + \Delta w_{ij}^k,$$

де індекси i та j визначають семантичний зв'язок, а індекс k — прошарок нейронів. Аналогічні корекції вводяться для порогових рівнів.

Для вихідного прошарку легко записати градієнт функції похибки по вагах як градієнт складної функції:

$$\frac{\partial E}{\partial w_{ij}^{last}} = \frac{\partial E}{\partial OUT_j^{last}} \frac{\partial OUT_j^{last}}{\partial NET_j^{last}} \frac{\partial NET_j^{last}}{\partial w_{ij}^{last}}. \quad (2)$$

Аналогічні розрахунки для порогів дадуть:

$$\frac{\partial E}{\partial q_j^{last}} = \frac{\partial E}{\partial OUT_j^{last}} \frac{\partial OUT_j^{last}}{\partial NET_j^{last}} \frac{\partial NET_j^{last}}{\partial q_j^{last}}. \quad (3)$$

Для вихідного прошарку матимемо:

$$\frac{\partial E}{\partial OUT_j^{last}} = (OUT_j^{last} - T \arg et_j^t). \quad (4)$$

Для похідної від зваженої суми ваг та для похідної від активаційної функції, відповідно, отримаємо:

$$\frac{\partial NET_j^{last}}{\partial w_{ij}^{last}} = \frac{\partial (\sum w_{ij}^{last} x_{ij}^{last} - q_j^{last})}{\partial w_{ij}^{last}} = x_{ij}^{last}, \quad (5)$$

$$\frac{\partial OUT_j^{last}}{\partial NET_j^{last}} = \frac{\partial F_j^{last}(NET_j^{last})}{\partial NET_j^{last}}. \quad (6)$$

Безпосередніми обчисленнями можна переконатись, що похідна по пороговому значенню дорівнює

$$\frac{\partial NET_j^{last}}{\partial q_j^{last}} = -1. \quad (7)$$

Для узагальнення формул (2) – (7) на довільний внутрішній прошарок нейронів, розглянемо наступне.

Для вихідного прошарку можемо записати:

$$\frac{\partial E}{\partial x_{ij}^k} = \frac{\partial E}{\partial OUT_j^k} \frac{\partial OUT_j^k}{\partial NET_j^k} \frac{\partial NET_j^k}{\partial x_{ij}^k},$$

Вирази $\frac{\partial E}{\partial OUT_j^k}$ та $\frac{\partial OUT_j^k}{\partial NET_j^k}$ обчислюються за формулами (4) і (6).

Для похідної від NET отримаємо

$$\frac{\partial NET_j^k}{\partial x_{ij}^k} = \frac{\partial (\sum w_{ij}^{k-1} x_{ij}^{k-1} - q_j^k)}{\partial x_{ij}^{k-1}} = w_{ij}^k.$$

Похідна по вхідних значеннях для поточного прошарку нейронів збігається із похідною по вхідних значенням попереднього прошарку, тому

$$\frac{\partial E}{\partial x_{ij}^k} = \frac{\partial E}{\partial OUT_i^{k-1}}.$$

Остання формула відображає рекурсивний перехід від поточного прошарку до попереднього і є аналогом формули (4) для внутрішніх прошарків нейронів.

Отже, отримали необхідні співвідношення для внутрішніх прошарків, не використовуючи повторних перерахунків вхідних значень. Обчислювальні затрати для обчислення значень градієнта функції помилки пропорційні до затрат на обчислення значень самої функції.

Слід відзначити, що запропонований підхід вигідно використовувати як допоміжний у тих алгоритмах, де потрібне обчислення градієнта.

Очевидна також можливість паралельних обчислень для параметрів одного і того ж прошарку нейронів.

Розглянутий метод виявився дуже привабливим. Його використання знайшло своє місце і в сучасності, а ідея лягла в основу великої кількості сучасних методів навчання. Важливими модифікаціями методу зворотного поширення помилки є методи вищих порядків. Зокрема, у праці [4] описано метод, який названо зворотним поширенням помилки другого порядку. Він використовує другі похідні для більш точної оцінки потрібної корекції параметрів ШНМ. Слід зазначити, що в [4] показано оптимальність запропонованого алгоритму в тому сенсі, що не можливо покращати оцінку, використовуючи похідні більш високого порядку.

Зрозуміло, що цей метод вимагає додаткових обчислювальних затрат порівняно із методом зворотного поширення помилки першого порядку, і потрібні дальші експерименти для оправдання цих затрат.

З погляду класичних методів оптимізації метод зворотного поширення помилки другого порядку є певним аналогом методу Ньютона, безпосереднє застосування якого неможливе з уже згаданих причин.

Для описаного підходу характерні такі проблеми.

- Для аналогів методу Ньютона потрібне “добре” початкове наближення.
- Описані методи не гарантують збіжності до глобального мінімуму. Тому при навчанні вони можуть “застрягнути” у локальному мінімумі функції помилки. Крок у цих методах вибирається не оптимально (практично це деяка константа менша одиниці). Поверхня функції помилки $E(P)$ має величезну кількість долин (локальних мінімумів).

Тому перший вибраний мінімум рідко має малу величину $E(P)$. Для покращання ситуації, деякі автори пропонують використовувати так зване навчання із шумом. Його суть полягає у додаванні до приросту параметрів значення деякої випадкової змінної із нульовим математичним сподіванням та маленькою дисперсією, тобто

$$\Delta w_{ij}^k = -\varepsilon \left(\frac{\partial E(W, Q)}{\partial w_{ij}^k} \right)_{w, Q} + \eta. \text{ В ролі } \eta \text{ часто вибирають змінні із розподілом}$$

Гаусса. Додавання шуму знижує швидкість навчання, але сприяє уникненню неглибоких локальних мінімумів.

Для навчання з учителем застосовують також *стохастичний підхід* [7]. Його можна описати такою процедурою.

1. Вибираємо параметри ШНМ випадковим чином і підкоректуємо їх на невелику випадкову величину. Подаємо множину входів і обчислюємо реакцію мережі.

2. Порівнюємо отриману реакцію із відповідним еталонним значенням і обчислюємо величину різниці між ними. Іншими словами, обчислюємо значення функції помилки.
3. Вибираємо випадково деякий параметр і підкоректуємо його на певну випадкову величину. Якщо відкоректований параметр сприяє зменшенню функції помилки, то його зберігаємо, інакше повертаємось до попереднього значення параметра.
4. Повторюємо кроки 1 — 3, доки ШНМ не буде навчена згідно з нашими вимогами.

Цей процес намагається мінімізувати $E(P)$, але може потрапити у пастку (рис. 1). Нехай спочатку параметри ШНМ підбрані так, що значення $E(P)$ попадає у точку А. Якщо випадкові кроки для параметрів є малими, то будь-які відхилення функції $E(P)$ будуть гіршими, а тому параметри будуть відкинуті. І $E(P)$ ніколи не потрапить у точку В.

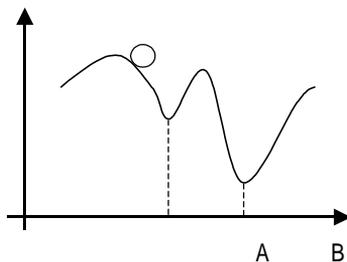


Рис. 1.

Якщо випадкові корекції параметрів є надто великими, то точки А та В будуть часто відвідуватись. Це буде справджуватися і для решти точок. Параметри будуть змінюватись так різко, що $E(P)$ ніколи не попаде у бажаний мінімум.

Корисна стратегія для уникнення такої проблеми полягає у виборі великих початкових кроків і поступовому їх зменшенні. Це дає змогу ШНМ вийти із локального мінімуму і водночас гарантує остаточну її стабілізацію.

Отже, стохастичні методи дають змогу частково вирішити проблему локальних мінімумів, яка властива методам із детерміністського класу. Як аналогію уявимо, що на рис. 1 зображено кульку на деякій поверхні в коробці. Якщо коробку сильно трясти у горизонтальному напрямку, то кулька буде швидко перекочуватися від одного краю до іншого. Тобто у кожен момент часу з рівною ймовірністю вона буде знаходитись у довільній точці поверхні.

Поступово зменшуючи силу трясіння, ми дійдемо до моменту, коли кулька буде на короткий час зупинятися у точці В. При дальшому зменшенні сили трясіння коробки досягнеться момент, коли воно є ще досить сильним щоб кулька перекотилася із точки А у точку В, але назад ні. У результаті ми маємо підстави сподіватись на збіжність до глобального мінімуму.

Саме так і навчаються ШНМ із використанням стохастичних методів. Популярними серед таких методів є Больцманівське навчання та навчання Коші. Їх відмінність полягає в тому, що в першому методі використовується розподіл Больцмана $P(c) = \exp(-c/kT)$

для визначення приймати чи ні новоутворені параметри, а у другому випадку для цієї мети використовується розподіл Коші $P(x) = \frac{T(t)}{T(t)^2 + x^2}$, де T — температурна функція.

Отже, застосування стохастичних методів дає змогу вирішити дві проблеми. По-перше, знайти глобальний мінімум, або, принаймні, його добре наближення, по-друге, забезпечити збіжність з довільного початкового наближення. Разом з тим стохастичні методи мають певні недоліки.

Довгий термін навчання (більш ніж у сто разів довше, порівняно із методом зворотного поширення помилки).

Висока ймовірність паралічу мережі — навчання продовжується, але саме навчання ШНМ припиняється або дуже сповільнюється. Це трапляється, коли параметри набувають дуже великих значень. Слід зазначити, що параліч, у першу чергу, стосується методу Коші. Розподіл Коші має нескінченну дисперсію і тому параметри можуть ставати як завгодно великими.

Генетичні алгоритми. Для навчання ШНМ можна використовувати еволюційні алгоритми, зокрема, генетичні (ГА). ГА належать до навчання з учителем, але вони не підпадають під означення детерміністського чи стохастичного підходів — це *еволюційний* підхід. ГА ґрунтуються на моделюванні розвитку біологічної популяції [3].

Нехай ми маємо вже описану функцію помилки $E(P)$. Популяцією назвемо набір параметрів P . Елемент P_i є представниками популяції. Елементи набору P можуть еволюціонувати за такими правилами:

1. Якщо $E(P_0)$ є малою величиною, то P_0 вважаємо вдалими параметрами. Ймовірність їх участі в еволюції збільшується, а ймовірність загинути зменшується.
2. Якщо $E(P_0)$ є великою величиною, то P_0 вважаємо невдалими параметрами. Ймовірність їх участі в еволюції зменшується, а ймовірність загинути збільшується.
3. Всі параметри мають однакову ймовірність мутації — зміщення на малу величину $P_0 \leftarrow P_0 + \Delta P$, де ΔP є невеликим за модулем вектором і характеризує величину мутації. Правило, за яким обчислюється ΔP , визначається конкретною реалізацією ГА. Найпростішою реалізацією ΔP може бути багатомірна випадкова змінна із нульовим математичним сподіванням.
4. Відповідно до ймовірностей, визначених на кроках 1 та 2, параметри розмножуються. Закони розмноження можуть бути різноманітними. Наприклад, дві близькі точки P_0 та P_1 , такі, що $\|P_0 - P_1\|$ є малою величиною, можуть утворити нового члена популяції за формулою $P_{new} = c_i(P_0 - P_1) + P_1 + m_i$, де c_i — скаляр, m_i — вектор, який залежить лише від i .
5. Згідно з ймовірностями, що визначені на кроках 1 та 2, параметри вмирають, тобто повністю вилучаються із набору параметрів P .

Поточний набір параметрів P_i є генофондом популяції.

Навчання ШНМ за допомогою ГА відбувається за такою схемою:

1. Створюємо певну кількість ШНМ (однакових за структурою) з різними значеннями P . Так ми утворюємо початкову популяцію.
2. Покроково імітується еволюція згідно з ГА.
3. На кожній ітерації вибирається деяке P_0 і обчислюється $E(P_0)$. Якщо значення достатньо мале, то припиняємо навчання.

Використання генетичних алгоритмів має такі переваги: збіжність до глобального мінімуму, відсутність проблеми вибору початкового наближення, абсолютний паралелізм при навчанні, біоподібність (ГА базуються на тих самих принципах, що й ШНМ, а це дає змогу природно використовувати ці алгоритми для навчання останніх).

На жаль, ГА не позбавлені недоліків. Вкажемо на основні.

- Кожен представник популяції (а це є конкретна ШНМ) вимагає багато пам'яті, оскільки кількість параметрів може бути дуже великою.
- Схильність до паралічу, оскільки визначення параметрів має псевдовипадковий характер, то вони можуть зростати довільно.
- Низька швидкість на фон-Нейманівських машинах. Мутації параметрів є випадковими, а тому вимагають великого обсягу обчислень. При відсутності можливості проведення паралельних обчислень це призводить до сильного зменшення ефективності алгоритмів.

Навчання без учителя. Доволі цікавою є парадигма навчання без учителя. У праці [5] подано корисні результати досліджень Кохонена щодо ШНМ, які здатні до самоорганізації. Розглянуті у цій праці ШНМ класифікує вхідні образи. Після навчання надання вхідного вектора із конкретного класу призводитиме до збудження конкретного нейрона, який і представляє класифікацію. Оскільки навчання відбувається без подання еталонної реакції, то наперед не відомо, який нейрон буде відповідати конкретному класу вхідних векторів. Але це можна буде визначити вже після навчання, подавши конкретні вектори на вхід ШНМ.

Запропонований Кохоненом алгоритм трактує набір з n ваг семантичних зв'язків нейрона як вектор у n -вимірному просторі. Перед навчанням кожній компоненті цього вектора присвоюється випадкова величина, а сам вектор ваг семантичних зв'язків нормалізується. Усі вхідні вектори також нормалізуються.

Навчання відбувається за таким алгоритмом.

1. Деякий вектор X подається на вхід ШНМ.
2. Визначається відстань D_j (в n -вимірному просторі) між X та ваговими векторами W_j кожного нейрона. Наприклад, у Евклідовому просторі вона може бути обчислена за формулою $D_j = \sqrt{\sum_i (x_i - w_{ij})^2}$, де x_i — компонента i вхідного вектора X , w_{ij} — вага входу i нейрона j .
3. Нейрон, ваговий вектор якого є найближчим до вхідного вектора, оголошується переможцем. Позначимо цей ваговий вектор через W_c і вважатимемо його основним у групі вагових векторів, які знаходяться на заданій нами відстані D від W_c .
4. Визначена таким чином група вагових векторів підкоректується за такою формулою

$$W_j(t+1) = W_j(t) + \alpha(X - W_j(t)),$$

де α — задана нами величина.

5. Повторюємо кроки 1 – 4 для всіх вхідних векторів.

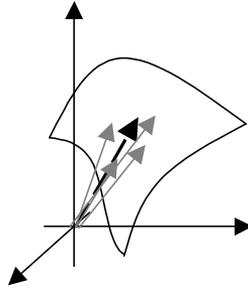


Рис. 2.

Протягом навчання ШНМ величини D та α поступово зменшуються. У [5] рекомендується покласти на початку навчання коефіцієнт α , який дорівнює 1 і зменшувати під час навчання до 0. D на початку навчання може дорівнювати максимальній відстані між ваговими векторами, а до кінця навчання має стати настільки маленьким, що буде навчатися лише один нейрон.

Точність класифікації буде покращуватися при додатковому навчанні. Для отримання хорошої точності кількість навчальних циклів (повторного навчання) має бути принаймні в 500 раз більша за кількість нейронів у вихідному прошарку.

Фактично, вхідні вектори під час навчання групуються в класи відповідно до їх розташування у просторі. Конкретний клас буде асоційовано із конкретним нейроном — його ваговий вектор буде відкоректований так, щоб займати геометрично центральне положення у класі (рис. 2).

При дальшому використанні навченої ШНМ ми визначатимемо до якого класу належить конкретний вхідний вектор.

Навчання з коректором — нова парадигма навчання ШНМ. Із викладеного можна зробити висновок, що існуюча теорія навчання ШНМ вимагає дальшого розвитку і пошуку нових парадигм, нових концепцій навчання.

Акцентуємо увагу на недоліках та перевагах існуючих підходів.

Дуже важливим є розвиток парадигми навчання без учителя, тобто без використання навчальної множини. Фактично цей підхід передбачає утворення ШНМ, яка здатна до самоорганізації. Прикладом є ШНМ Кохонена [5]. Суть самоорганізації у ШНМ Кохонена полягає в групуванні вхідних векторів у класи. Це досягається за допомогою такого налаштування параметрів ШНМ, що близькі вхідні вектори активізують один і той самий нейрон ШНМ Кохонена. Якщо на вхід подається вектор, який не підходить до жодного класу, то додається новий нейрон (чи група нейронів), який би реагував на новий клас. У цих мережах важливий сам факт реагування нейронів. А отже значення реакцій реагуючих нейронів є нуль чи одиниця.

ШНМ Кохонена є чи не єдиним прикладом навчання без учителя і ґрунтується на механізмі *класифікації* вхідних образів.

Навчання з учителем знайшло широке застосування для вирішення низки прикладних проблем. Основна проблема цього підходу полягає у скінченності навчальної множини. Навіть при існуванні методу навчання, який би знаходив глобальний мінімум функції помилки, навчена ШНМ не може гарантувати коректної реакції на образ, який є достатньо далекий від навчальної множини. А при великому відхиленні вхідного образу від навчальної множини результати можуть бути взагалі неадекватні.

Автори пропонують нову парадигму, яка є певним синтезом навчання із учителем і без нього. Цей підхід названо *навчанням із коректором*. В його основу покладено такі принципи:

- відсутність навчальної множини (без чого не може бути реалізоване навчання з учителем);
- врахування величини реакції нейронів;
- наявність механізму корекції, який би згідно з певними правилами визначав “правильність” отриманої реакції ШНМ.

Відсутність навчальної множини дає змогу зняти відзначені вище обмеження на ШНМ. Це сприяє розширенню класу задач, для яких може бути застосована ШНМ.

Відмова від навчальної множини передбачає розробку механізму корекції (коректора), який враховує значення величини реакції ШНМ (чого не роблять ШНМ Кохонена). Коректор складається із двох підсистем: перша визначає придатність поточної реакції ШНМ, друга — обчислює деякий коефіцієнт корекції. Друга підсистема вступатиме у дію лише тоді, коли перша підсистема виявить непридатність поточної реакції ШНМ. Коефіцієнт, вироблений другою підсистемою, використовується навчальним алгоритмом для корегування параметрів ШНМ.

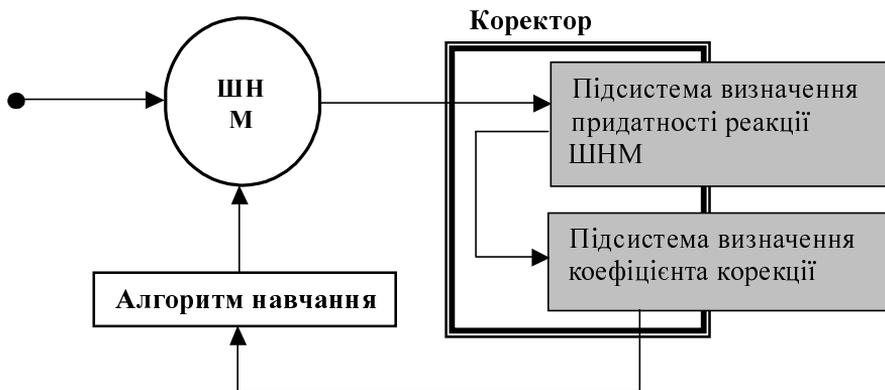


Рис. 3

Коректор знімає потребу у навчальній множині і певною мірою регулює процес самоорганізацію. Водночас коректор не перешкоджає використанню відомих методів навчання. Зокрема для коректуючого коефіцієнта можна провести певну аналогію із методом зворотного поширення помилки, в якому при навчанні вихідного прошарку використовується різниця між реакцією ШНМ та відповідним еталонним значенням. Замість цієї різниці можна використати коефіцієнт корекції. Якщо для навчання буде використовуватися стохастичний чи еволюційний підхід, то підсистема визначення коефіцієнта корекції не буде задіяна. Укрупнена схема реалізації підходу навчання з коректором зображена на рис. 3.

На закінчення покажемо, як можна використати коректор для вдосконалення ШНМ Кохонена. Нагадаємо, що ця ШНМ займається класифікацією вхідних векторів — кожному класу векторів виділяється окремий нейрон, реакція якого свідчить, що вхідний вектор належить саме до цього класу.

Вважатимемо, що вхідні вектори належать до одного класу, якщо відстань між ними не перевищує деякого числа D_j , де j — це номер класу.

Підсистема визначення придатності реакції ШНМ працює за таким алгоритмом.

Ініціалізація

- 1) Подаємо довільний вектор (параметрів, чинників впливу тощо) A^1 на вхід ШНМ. Отриману реакцію ШНМ вважаємо правильною і покладаємо *еталоном* класу. Тим самим започатковуємо *перший* клас. Сам вектор назвемо *породжувальним* вектором цього класу. $N \leftarrow 1$.

Основний цикл

- 2) Подаємо на вхід поточний вектор B . Якщо відстань від вектора B до кожного з породжувальних векторів $A^1 \cdots A^N$ є великою, то переходимо на крок 4. Інакше — на крок 3.
- 3) Вектор B належить до одного з утворених класів і далі діємо залежно від реакції ШНМ на векторі B . Якщо вона є далекою від еталону класу, якому належить вектор B , то коректор вважає, що ШНМ потребує навчання. Включається в дію підсистема обчислення коефіцієнта корекції. Активізується алгоритм навчання (він використовує обчислений коефіцієнт корекції). Повторюємо крок 3. Якщо ж реакція ШНМ була близькою, то переходимо на крок 2.
- 4) Якщо реакція ШНМ на векторі B була близькою хоча б до однієї з реакцій ШНМ на векторах $A^1 \cdots A^N$, то коректор вважає, що ШНМ потребує навчання. Включається в дію підсистема обчислення коефіцієнта корекції. Активізується алгоритм навчання (він використовує обчислений коефіцієнт корекції). Повторюємо крок 4. Якщо ж реакція була далекою, то започатковуємо новий клас і вектор B оголошується породжувальним вектором цього класу: $A^{N+1} \leftarrow B$. Реакція вважається *еталоном* для новоутвореного класу. $N \leftarrow N + 1$. Переходимо на крок 2.

Слід відзначити, що при навчанні з коректором описаний алгоритм діє упродовж всього часу функціонування ШНМ. Таким чином ШНМ навчається весь час своєї роботи, тобто життя.

Підсистема визначення коефіцієнта корегування може бути реалізована по-різному. Зокрема, за коректуючий коефіцієнт можна приймати різницю між еталонною реакцією ШНМ для конкретного класу та поточною реакцією ШНМ.

Результат роботи ШНМ з коректором схематично показаний на рис. 4.

Класифікаційні властивості запропонованого підходу до навчання очевидні. Перевага перед ШНМ Кохонена полягає у тому, що отримуються числові характеристики кожного вхідного вектора у кожному класі. Покажемо як цей факт може бути використаний.

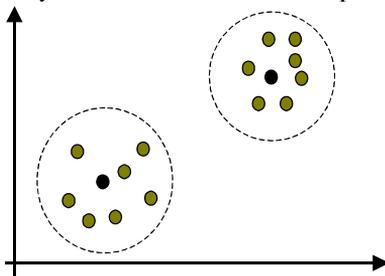


Рис. 4.

Нехай потрібно кількісно охарактеризувати деяке явище, яке залежить від певних параметрів (вони утворюватимуть вхідні вектори для ШНМ). Протягом певного часу ШНМ

буде просто навчатися. Потім для кожного класу можна встановити, чи він відповідає доброму стану речей, чи поганому (згідно із реальною ситуацією). Таким чином ми отримуємо “негативні” та “позитивні” класи, а значення реакцій у цих класах можуть бути чисельним вираженням “негативності” чи “позитивності”.

1. Грицик В. Розробка інформаційних технологій функціонування, програмування і налаштування нейронних систем паралельної обробки сигналів// Інформаційні технології і систем.. – 1998. – № 1/2. – С. 9 – 14. 2. Грицик В., Азенберг Н., Бунь Р., Данилюк О., Гече Ф., Кисіль Б., Олексів Б., Опотяк Ю., Стрямець С., Ткаченко Р., Вальковський В., Войчишин К. Нейронні та нейроподібні мережі: синтез, реалізація, застосування та майбутнє// Інформаційні технології і системи. – 1998. – № 1/2. – С. 15 – 55. 3. Нікольський Ю., Щербина Ю. Генетичні алгоритми в екстремальних задачах// Вісн. Львівського університету. – 2000. – Сер. прикладна математика та інформатика. – Вип. 2. – С. 197 – 208. 4. Parker D. Second order back propagation: Implementation an optimal $O(n)$ approximation to Newton's methods as an artificial neural networks. Manuscript submitted for publication, 1987. 5. Kohonen T. Self-organization and associative memory. Series in Information Sciences. Vol. 8. verlag, 1984. 6. Rumelhart D., Hinton G., Williams R. Learning internal representations by error propagation. In Parallel distributed processing, Cambridge, MA: MIT Press. – 1986. – Vol. 1. – P. 318 – 62. 7. Geman S., Geman D. Stochastic relaxation, Gibbs distribution and Baysian restoration of images// IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1984. – 6. – P. 721 – 41.

УДК 681.3

Якушев В.В.

ТЗОВ “СофтСерв-Консалтинг”

ПЕРЕТВОРЕННЯ І ОПРАЦЮВАННЯ ВХІДНИХ ПОТОКІВ ДАНИХ З ВИКОРИСТАННЯМ ШВИДКОЗБІЖНИХ РЯДІВ

© Якушев В.В., 2001

Mathematical algorithms for using in information technology are considered.

Тут описані математичні алгоритми, які використовуються у інформаційних технологіях

Перетворення і опрацювання вхідних потоків даних є одним з основних етапів в процесі їх підготовки для інформаційного наповнення баз даних та знань .

У загальному вигляді перетворенням можна вважати відображення фізичної величини, що перетворюється, на деякий швидкозбіжний ряд, кількість якої виражається границею фундаментальної послідовності раціональних чисел [1]. На практиці таке перетворення вхідних потоків даних (фізичних величин) може бути реалізоване за допомогою аналого-цифрового перетворення . Оскільки характеристики реальних аналого-цифрових перетворювачів (АЦП) мають технічно і технологічно обумовлені обмеження, дослідження їх впливу на результати перетворення мають теоретичне і практичне значення. Одним з таких обмежень є скінченна чутливість АЦП, іншими словами всі значення величини, що менші або рівні порогу чутливості АЦП, сприймаються як рівні нулю. Отже кількість величини, що перетворюється, теоретично дорівнює повній сумі ряду, а практично, точність наближення визначається кількістю врахованих членів ряду.

В роботі досліджуються перетворення вхідних потоків даних на прикладі швидкозбіжних рядів Остроградського и комбінованого [2]. Оцінюється похибка перетворення фізичної