УДК 519.7:004.8

Бодянський Є. В., Тесленко Н. О., Дейнеко А. О.

ЕВОЛЮЦІЙНА НЕЙРОННА МЕРЕЖА З ЯДЕРНИМИ ФУНКЦІЯМИ АКТИВАЦІЇ Й АДАПТИВНИЙ АЛГОРИТМ ЇЇ НАВЧАННЯ

Запропоновано архітектуру еволюційної нейронної мережі, що не схильна до прокляття розмірності, здатна обробляти інформацію в режимі реального часу, адаптуючи при цьому свої параметри і структуру до умов задачі. У якості активаційних функцій було використано ядерні функції.

Ключові слова: *еволюційна нейронна мережа, радіально-базисна нейронна мережа, узагальнена регресійна нейронна мережа, машина опорних векторів, ядерна функція активації.*

Предложена архитектура эволюционной нейронной сети, которая не склонна к проклятию размерности, способна обрабатывать информацию в режиме реального времени, адаптируя при этом свои параметры и структуру к условиям задачи. В качестве активационных функций были использованы ядерные функции активации.

Ключевые слова: *эволюционная нейронная сеть, радиально-базисная нейронная сеть, обобщенная регрессионная нейронная сеть, машина опорных векторов, ядерная функция.*

In this work the evolving artificial neural network with kernel activating function that contains support vector machine, radial basis function neural network and general regression network as subsystems is proposed. This network is tuned using both optimization based and memory based approaches and inclined to the «curse of dimensionality», is able to real time mode information processing by adapting its parameters and structure to problem conditions.

Key words: *evolving neural network radial-basis neural network, general regression network, support vector machine, kernel activating function.*

Вступ

Для розв'язання широкого класу задач обробки інформації, ідентифікації систем і об'єктів керування, насамперед, суттєво нелінійних та функціонуючих в умовах структурної й параметричної невизначеності, широке розповсюдження отримали штучні нейронні мережі (ШНМ), завдяки, насамперед, універсальним апроксимуючим властивостям і здатності до навчання. При цьому, кажучи про навчання, слід ураховувати можливість як настроювання синаптичних ваг і параметрів активаційних функцій, так і архітектури мережі в цілому. Ці властивості характерні для досить нового класу ШНМ – еволюційних коннекціоністських систем [1], що мають низку переваг перед традиційними нейромережами.

В ситуаціях, коли необхідна обробка інформації в on-line режимі по мірі послідовного надходження на вхід системи нових даних, на перший план виходить питання швидкості процесу навчання, що суттєво обмежує клас ШНМ, придатних для роботи в цьому режимі. З погляду оптимізації, по швидкості процесу навчання досить перспективними є ШНМ, що базуються на ядерних (радіально-базисних, потенційних, дзвонуватих) функціях активації, найбільш характерними представниками яких є радіально-базисні нейронні мережі (RBFN), узагальнені регресійні нейронні мережі (GRNN) та машини опорних векторів (SVM) [2].

Найбільш популярними з перелічених ШНМ є радіально-базисні нейронні мережі, вхідний (нульовий) шар яких утворений сенсорами, що зв'язують мережу з навколишнім середовищем. Прихований шар реалізує деяке нелінійне перетворення простору входів R^n у прихований простір R^{h_R} більш високої розмірності ($h_R > n$), а вихідний шар, утворений адаптивним лінійним асоціатором, формує відгук мережі $\hat{y}^R$ на вхідний сигнал $x = (x_1, x_2, ..., x_n)^T$. Властивості RBFN визначаються радіально-базисними функціями $\varphi(x) = \varphi(\|x - c\|, \sigma)$, що залежать від відстані $\|x - c\|$ між вхідним вектором x і центром c, а також параметру масштабу σ, що визначає локальну область вхідного простору, на яку реагує ця функція. Найчастіше в якості активаційної функції застосовується ґавсіан.

$$\varphi_l(x) = \exp\left(-\frac{\|x - c_l\|^2}{2\sigma^2}\right), \ l = 1, 2, ..., h_R,$$

хоча може бути використана й множина інших подібних конструкцій.

Навчання RBFN ґрунтується на оптимізації деякого (зазвичай квадратичного) критерію, а оскільки вихідний сигнал $\hat{y}^R$ лінійно залежить від синаптичних ваг, для їхнього настроювання можуть бути використані швидкодіючі алгоритми оптимізації другого порядку, наприклад, метод найменших квадратів.

Основним недоліком радіально-базисної мережі є її схильність до «прокльону розмірності», що спричиняє експоненційне зростання кількості радіально-базисних нейронів (R-нейронів) із зростанням розмірності вхідного простору.

Ефективною альтернативою ШНМ, навчання яких базується на оптимізації, є мережі з навчанням, заснованим на пам'яті, типовим представником яких є узагальнена регресійна нейронна мережа (GRNN). Навчання GRNN зводиться до установлення центрів активаційних функцій (зазвичай тих же ґавсіанів) у точках вхідного простору з координатами, що визначаються векторами даних, які надходять на обробку. Таким чином, GRNN й інші подібні їй мережі навчаються практично миттєво. Особливістю цієї мережі є те, що кількість нейронів прихованого шару h_G заздалегідь не фіксується й змінюється в процесі навчання. У такий спосіб мережа еволюціонує в часі, поліпшуючи свої характеристики.

Основним недоліком узагальненої регресійної нейронної мережі є чутливість до збурень та перешкод, оскільки в загальному випадку GRNN вирішує задачу не апроксимації, а інтерполяції.

Своєрідним гібридом RBFN і GRNN є машина опорних векторів (SVM), що об'єднує в собі навчання, яке базується як на оптимізації, так і на пам'яті, а також реалізує метод мінімізації емпіричного ризику. Ключовим моментом при синтезі цієї мережі є опорні вектори, що становлять собою невелику підмножину найбільш інформативних векторів даних, виділених у процесі навчання. Маючи архітектуру, що збігається з RBFN і GRNN, машини опорних векторів є найбільш ефективними нейромережами в умовах малих вибірок даних, що забезпечують високу якість апроксимації.

Основним недоліком звичайних SVM є чисельна громіздкість процедури визначення синаптичних ваг, що зводиться до задачі нелінійного програмування з обмеженнями-нерівностями, кількість яких h_{SV} визначається об'ємом оброблюваної вибірки. Більш ефективними з цього погляду є машини опорних векторів, що базуються на методі найменших квадратів (LS-SVM) [3], та навчаються за допомогою методів квадратичного програмування, однак так чи інакше це навчання відбувається в пакетному режимі.

Метою цієї роботи є синтез адаптивних алгоритмів навчання для комбінованої еволюційної нейронної мережі з ядерними функціями активації, що поєднує в собі переваги RBFN, GRNN та SVM, та дозволяє обробляти дані в режимі реального часу по мірі їх надходження.

1. Архітектура еволюційної штучної нейронної мережі

Архітектура запропонованої нейронної мережі наведена на рис. 1 і складається з трьох паралельно підключених до входів (нульового шару) однотипних підсистем: RBFN, GRNN і SVM [4].

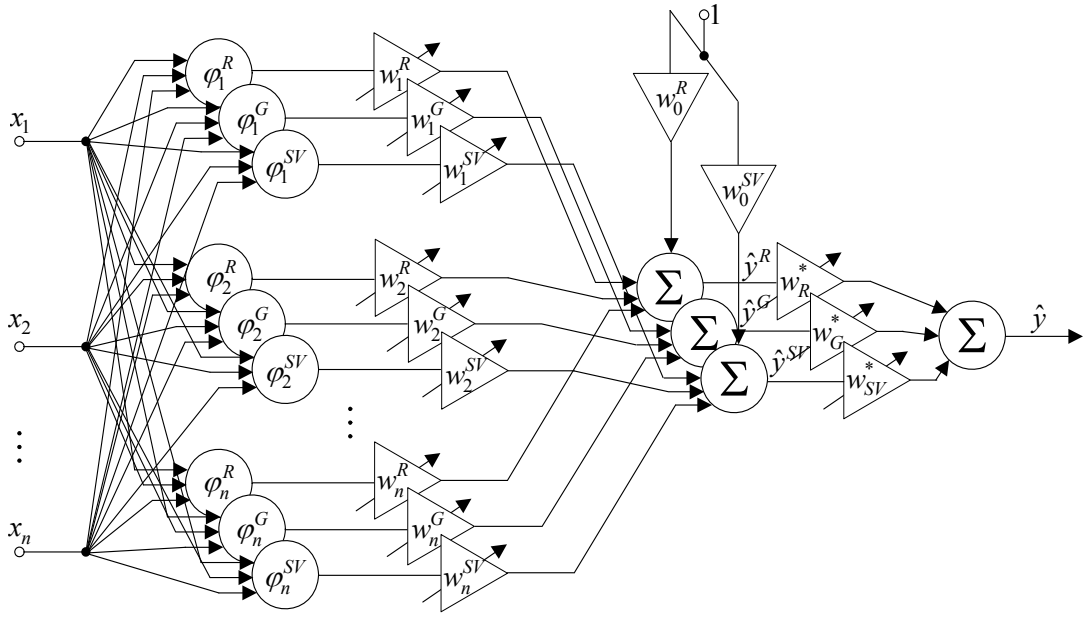

Рис. 1. Еволюційна штучна нейронна мережа

На нульовий шар подається векторний сигнал $x(k) = (x_1(k), x_2(k), ..., x_n(k))^T \in R^n$ (тут $k = 1, 2, ...$ – індекс поточного дискретного часу), який далі обробляється однотипними R-нейронами з активаційними

функціями $\varphi_l^R(x(k))$, $\varphi_l^G(x(k))$, $\varphi_l^{SV}(x(k))$, де індекси R, G, SV означають, що відповідний нейрон входить до складу RBFN, GRNN і SVM. Сигнали з виходів R-нейронів надходять на входи трьох лінійних асоціаторів, кожний з яких містить набір синаптичних ваг, що описуються векторами w^R, w^G і w^{SV}, які підлягають визначенню в процесі навчання. Виходами цих лінійних асоціаторів є скалярні сигнали $\hat{y}^R(k)$, $\hat{y}^G(k)$ та $\hat{y}^{SV}(k)$. Порівняння цих виходів із зовнішнім навчальним сигналом $y(k)$ дозволяє формувати помилки навчання $e^R(k)$, $e^G(k)$ та $e^{SV}(k)$, на основі яких визначаються w^R, w^G і w^{SV}.

Вихідний шар утворений адаптивним лінійним асоціатором із трьома входами й скалярними синаптичними вагами w_R^*, w_G^* і w_{SV}^*. Тут проводиться аналіз сигналів $\hat{y}^R(k)$, $\hat{y}^G(k)$ та $\hat{y}^{SV}(k)$ й синтезується оптимальний вихід мережі $\hat{y}^*(k)$ на основі похибки навчання $e^*(k) = y(k) - \hat{y}^*(k)$.

2. Адаптивні алгоритми навчання

Процес навчання еволюційної нейронної мережі умовно можна розділити на чотири незалежні процедури: настроювання RBFN, настроювання GRNN, настроювання SVM і настроювання вихідного лінійного асоціатора, причому кожна із цих процедур повинна мати або рекурентну форму, або інший вигляд, що допускає реалізацію в реальному часі.

Перетворення, що реалізується радіально-базисной нейронною мережею, може бути записане у вигляді

$$\hat{y}^R(k) = w_0^R + \sum_{l=1}^{h_R} w_e^R \varphi_l^R(x(k)) = \sum_{l=0}^{h_R} w_l^R \varphi_l^R(x(k)) = w^{RT}\tilde{\varphi}^R(x(k)),$$

де $\tilde{\varphi}^R(x(k)) = (1, \varphi^{RT}(x(k)))^T$, $\varphi^R(x(k)) = (\varphi_1^R(x(k)), \varphi_2^R(x(k)), ..., \varphi_{h_R}^R(x(k)))^T$,

T – символ транспонування.

Оптимізація квадратичного критерію навчання

$$E^R(k) = \frac{1}{2}\sum_k (y(k) - \sum_{l=0}^{h_R} w_l^R \varphi_l^R(x(k)))^2 = \frac{1}{2}\sum_K (y(k) - w^{RT}\tilde{\varphi}^R(x(k)))^2$$

у реальному часі може бути реалізована за допомогою рекурентного методу найменших квадратів у вигляді

$$\begin{cases} w^R(k+1) = w^R(k) + \dfrac{P^R(k)(y(k+1) - w^{RT}(k)\tilde{\varphi}^R(x(k+1)))}{1 + \tilde{\varphi}^{RT}(x(k+1))P^R(k)\tilde{\varphi}^R(x(k+1))}\tilde{\varphi}^R(x(k+1)), \\ P^R(k+1) = P^R(k) - \dfrac{P^R(k)\tilde{\varphi}^R(x(k+1))\tilde{\varphi}^{RT}(x(k+1))P^R(k)}{1 + \tilde{\varphi}^{RT}(x(k+1))P^R(k)\tilde{\varphi}^R(x(k+1))}. \end{cases}$$

Оскільки навчання узагальненої регресійної мережі відбувається миттєво за принципом «нейрони в точках даних», відгук GRNN на довільний сигнал x має вигляд

$$\hat{y}^G(x) = \frac{\sum_{l=1}^{h_G} y(l)\varphi_l^G(x)}{\sum_{l=1}^{h_G} \varphi_l^G(x)} = \frac{\sum_k y(k)\exp\left(-\dfrac{\|x - x(k)\|^2}{2\sigma^2}\right)}{\sum_k \exp\left(-\dfrac{\|x - x(k)\|^2}{2\sigma^2}\right)} = \frac{NG(k)}{DG(k)},$$

де центри активаційних функцій $\varphi_l^G(x)$ визначаються всією навчальною вибіркою $x(1)$, ..., $x(k)$, при цьому зрозуміло, що в загальному випадку $h_G = k$.

Якщо ж дані надходять на обробку послідовно, уточнення реакції мережі з урахуванням нових спостережень $x(k + 1)$, $y(k + 1)$ проводиться згідно із простим співвідношенням

$$\hat{y}^G(x) = \frac{\sum_{l=1}^{k+1} y(l)\varphi_l^G(x)}{\sum_{l=1}^{k+1} \varphi_l^G(x)} = \frac{NG(k) + y(k+1)\exp\left(-\dfrac{\|x - x(k+1)\|^2}{2\sigma^2}\right)}{DG(k) + \exp\left(-\dfrac{\|x - x(k+1)\|^2}{2\sigma^2}\right)}.$$

Перетворення, що реалізується машиною опорних векторів, може бути записане у вигляді

$$\hat{y}^{SV}(x) = w_x^{SVT}\varphi^{SV}(x) + w_0^{SV}, \tag{1}$$

де $w_x^{SV} = (w_1^{SV}, ..., w_l^{SV}, ..., w_{h_{SV}}^{SV})^T$, $\varphi^{SV}(x) = (\varphi_1^{SV}(x), ..., \varphi_{h_{SV}}^{SV}(x))^T$, а її навчання (у випадку LS-SVM) зводиться до одночасного встановлення центрів активаційних функцій у точках навчальної вибірки $x(l)$, $l = 1, 2, ..., k$ подібно GRNN й оптимізації квадратичного критерію

$$E^{SV}(k) = \frac{1}{2}w_x^{SVT}w_x^{SV} + \frac{\gamma}{2}\sum_k e^2(k)$$

за наявності системи з $h_{SV} = k$ лінійних обмежень-рівностей:

$$\begin{cases} y(1) = w_x^{SVT}\varphi^{SV}(x(1)) + w_0^{SV} + e(1), \\ \vdots \\ y(k) = w_x^{SVT}\varphi^{SV}(x(k)) + w_0^{SV} + e(k), \end{cases}$$

де $\gamma > 0$ – параметр регуляризації,

$$e(l) = y(l) - \hat{y}^{SV}(x(l)).$$

У пакетному режимі задача навчання SVM зводиться до знаходження сідлової точки функції Лагранжа

$$L(w_x^{SV}, w_0^{SV}, e(k), \lambda(k)) = E^{SV}(k) + \sum_k \lambda(k)(y(k) - w_x^{SVT}\varphi^{SV}(x(k)) - w_0^{SV} - e(k)) =$$

$$= \frac{1}{2} w_x^{SVT} w_x^{SV} + \frac{\gamma}{2}\sum_k e^2(k) + \sum_k \lambda(k)(y(k) - w_x^{SVT}\varphi^{SV}(x(k)) - w_0^{SV} - e(k)), \tag{2}$$

при цьому крім власне синаптичних ваг w_x^{SV} і w_0^{SV} повинні бути знайдені й k невизначених множників Лагранжа $\lambda(l)$, $l = 1, ..., k$.

Система рівнянь Куна-Таккера для лагранжіана (2) може бути записана у вигляді

$$\begin{cases} \nabla_{w_x^{SV}} L = w_x^{SV} - \sum_k \lambda(k)\varphi^{SV}(x(k)) = \vec{0}_k, \\ \dfrac{\partial L}{\partial w_0^{SV}} = -\sum_k \lambda(k) = 0, \\ \dfrac{\partial L}{\partial e(l)} = \gamma e(l) - \lambda(l) = 0, \\ \dfrac{\partial L}{\partial \lambda(l)} = y(l) - w_x^{SVT}\varphi(x(l)) - w_0^{SV} - e(l) = 0 \end{cases}$$

(тут $\vec{0}_k$ – $(k \times 1)$ – вектор, утворений нулями) або

$$\begin{cases} w_x^{SV} = \sum_k \lambda(k)\varphi^{SV}(x(k)), \\ \sum_k \lambda(k) = 0, \\ \lambda(l) = \gamma e(l), \\ y(l) - w_x^{SVT}\varphi(x(l)) - w_0^{SV} - e(l) = 0. \end{cases} \tag{3}$$

З першого рівняння системи (3) випливає, що синаптичні ваги цілком визначаються значеннями невизначених множників Лагранжа, у зв'язку з чим навчання SVM зводиться до їхнього визначення, при цьому система (3) може бути переписана в компактному вигляді

$$\begin{pmatrix} 0 & I_k^T \\ I_k & \Omega(k) + \gamma^{-1} I_{k,k} \end{pmatrix} \begin{pmatrix} w_0^{SV} \\ \Lambda(k) \end{pmatrix} = \begin{pmatrix} 0 \\ Y(k) \end{pmatrix}, \tag{4}$$

де $\Lambda(k) = (\lambda(1), ..., \lambda(l), ..., \lambda(k))^T$, $I_k - (k \times 1)$ – вектор, утворений одиницями, $I_{k,k} - (k \times k)$ – одинична матриця,

$$Y(k) = (y(1), ..., y(l), ..., y(k))^T, \; i = 1, 2, ..., k, \; j = 1, 2, ..., k$$

$\Omega(k) = \left\{\Omega_{ij} = \varphi^{SVT}(x(i))\varphi^{SV}(x(j)) = K(x(i), x(j))\right\}$, $K = (x\ (i), x\ (j))$ – деяка ядерна функція, що задовільняє умовам теореми Мерсера [2], найчастіше той же ґавсіан

$$K(x(i), x(j)) = \exp\left(-\frac{\|x(i) - x(j)\|^2}{2\sigma^2}\right).$$

При цьому перетворення (1), що реалізується машиною опорних векторів, може бути переписане у формі

$$\hat{y}^{SV}(x) = \sum_k \lambda(k) K(x, x(k)) + w_0^{SV},$$

а його параметри $\lambda(k)$, w_0^{SV} можуть бути знайдені безпосередньо з (4) у вигляді

$$\begin{pmatrix} w_0^{SV} \\ \Lambda(k) \end{pmatrix} = \begin{pmatrix} 0 & I_k^T \\ I_k & \Omega(k) + \gamma^{-1} I_{k,k} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Y(k) \end{pmatrix} = P^{SV}(k) \begin{pmatrix} 0 \\ Y(k) \end{pmatrix}. \tag{5}$$

Зрозуміло, що адаптивне навчання SVM може бути організоване на основі чисельно простої процедури обернення матриці в правій частині системи (5).

Переписуючи (5) для $(k + 1)$-го моменту часу у вигляді

$$\begin{pmatrix} w_0^{SV} \\ \Lambda(k+1) \end{pmatrix} = \begin{pmatrix} 0 & I_{k+1}^T \\ I_k & \Omega(k+1) + \gamma^{-1} I_{k+1,k+1} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Y(k+1) \end{pmatrix} =$$

$$=\begin{pmatrix} & & 1 \\ 0 & I_k^T & K(x(1),x(k+1)) \\ I_k & \Omega(k)+\gamma^{-1}I_{k,k} & K(x(2),x(k+1)) \\ & & \vdots \\ & & K(x(k),x(k+1)) \\ 1K(x(1),x(k+1))...K(x(k),x(k+1)) & & 1+\gamma^{-1} \end{pmatrix}=\begin{pmatrix} 0 \\ Y(k) \\ y(k+1) \end{pmatrix}=$$

$$=\begin{pmatrix} (P^{SV}(k))^{-1} & \vec{K}(x(i),x(k+1)) \\ \vec{K}^T(x(i),x(k+1)) & 1+\gamma^{-1} \end{pmatrix}^{-1}\begin{pmatrix} \vec{Y}(k) \\ y(k+1) \end{pmatrix}$$

(тут $\vec{K}^T(x(i),\ x(k+1))=(1,K(x(1),x(k+1)),...,K(x(k),x(k+1)))^T$, $\vec{Y}^T(k)=(0,Y^T(k))$) і застосовуючи формулу Фробеніуса обернення блокових матриць [5], приходимо до простого виразу для розрахунків матриці $P^{SV}(k+1)$:

$$P^{SV}(k+1)=\begin{pmatrix} P^{SV}(k)+(P^{SV}(k)\vec{K}(x(i), & -(P^{SV}(k)\vec{K}(x(i),x(k+1))) \\ x(k+1))\vec{K}^T(x(i),x(k+1))P^{SV}) & (1+\gamma^{-1}-\vec{K}^T(x(i), \\ (1+\gamma^{-1}-\vec{K}^T(x(i),x(k+1)) & x(k+1))P^{SV}(k)\vec{K}(x(i), \\ P^{SV}(k)\vec{K}(x(i),x(k+1)))^{-1} & x(k+1)))^{-1} \\ & \\ -(\vec{K}^T(x(i),x(k+1))P^{SV}(k)) & (1+\gamma^{-1}-\vec{K}^T(x(k), \\ (1+\gamma^{-1}-\vec{K}^T(x(i),x(k+1)) & x(k+1))P^{SV}(k)\vec{K}(x(i), \\ P^{SV}(k)\vec{K}(x(i),x(k+1)))^{-1} & x(k+1)))^{-1} \end{pmatrix} \quad (6)$$

Зрозуміло, що при великих обсягах навчальної вибірки, обернення ($k \times k$)-матриць набагато зручніше робити, використовуючи формулу (6).

3. Навчання вихідного шару

Кожна з розглянутих тут ШНМ має свої переваги та недоліки, й при обробці реальних даних заздалегідь неможливо передбачити, яка з нейронних мереж буде найбільш ефективною. У зв'язку з цим у розглянуту еволюційну нейронну мережу введемо додатковий (вихідний) шар обробки інформації, що синтезує оптимальний вихідний сигнал $\hat{y}^*(k)$ на основі принципу адаптивного узагальненого прогнозування [6-8].

Вводячи узагальнені вектори вихідних сигналів $\hat{y}(k)=(\hat{y}^R(k),\hat{y}^G(k),\hat{y}^{SV}(k))^T$ і параметрів узагальнення у вигляді синаптичих ваг адаптивного лінійного асоціатора $w^*=(w_R^*,w_G^*,w_{SV}^*)^T$, знайдемо оптимальний відгук мережі у формі

$$\hat{y}^*(k)=w^{*T}\hat{y}(k)$$

при обмеженнях на зміщеність

$$I_3^T w^*=1.$$

Результат оптимізації функції Лагранжа

$$L(w^*,\lambda^*)=\frac{1}{2}\sum_k(e^*(k))^2+\lambda^*(I_3^T w^*-1)$$

веде до оцінки

$$\begin{cases} w^*(k)=w(k)+P^*(k)\dfrac{1-I_3^T w(k)}{I_3^T P^*(k)I_3}I_3, \\ w(k)=P^*(k)\sum_k\hat{y}(k)y(k)=\left(\sum_k\hat{y}(k)\hat{y}^T(k)\right)^{-1}\sum_k\hat{y}(k)y(k), \end{cases}$$

яка може бути переписана в рекурентній формі

$$\begin{cases} w^*(k+1)=w(k+1)+P^*(k+1)\dfrac{1-I_3^T w(k+1)}{I_3^T P^*(k+1)I_3}I_3, \\ w(k+1)=w(k)+\dfrac{P^*(k)(y(k+1)-w^T(k)\hat{y}(k+1))}{1+\hat{y}^T(k+1)P^*(k)\hat{y}(k+1)}\hat{y}(k+1), \\ P^*(k+1)=P^*(k)-\dfrac{P^*(k)\hat{y}(k+1)\hat{y}^T(k+1)P^*(k)}{1+\hat{y}^T(k+1)P^*(k)\hat{y}(k+1)}. \end{cases}$$

У такий спосіб реалізуються адаптивні процедури навчання RBFN, GRNN, SVM і вихідного шару еволюційної нейронної мережі.

Висновок

Розглянута й вирішена задача адаптивного навчання еволюційної нейронної мережі з ядерними функціями активації, що базуються на об'єднанні різних архетипів навчання (оптимізація, «нейрони в точках даних», комбіноване настроювання). Введені чисельно прості й швидкодіючі алгоритми навчання, що дозволяють у реальному часі синтезувати оптимальний відгук мережі. Цей підхід забезпечує розв'язання задач апроксимації, інтерполяції, прогнозування, ідентифікації, інтелектуального керування й т. п. у системах, коли інформація надходить на обробку послідовно в on-line режимі.

ЛІТЕРАТУРА

1. Kasabov N. Evolving Connectionist Systems / N. Kasabov. – London ; Springer ; Verlag, 2003. – P. 307.
2. Haykin S. Neural Networks. A Comprehensive Foundation. / S. Haykin. – Upper Saddle River ; N. J. : Prentice Hall, Inc., 1999. – 842 p.
3. Suykens J. A. K. Least Squares Support Vector Machines / J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, J. Vandewalle. – Singapore : World Scientific, 2002. – 294 p.
4. Бодянский Е. В. Комбинированные эволюционные нейронные сети с ядерными функциями активации в задачах идентификации / Е. В. Бодянский, Н. А. Тесленко, А. А. Дейнеко // Сб. тр. Междунар. научн. конф. «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта». – Евпатория, 2010. – С. 274–279.
5. Гантмахер Ф. Р. Теория матриц / Ф. Р. Гантмахер. – Москва : Наука, 1988 – С. 552.
7. Бодянский Е. В. Адаптивное обобщенное прогнозирование многомерных случайных последовательностей / Е. В. Бодянский, И. П. Плисс, Т. В. Соловьева // Доклады АН УССР. – 1989. – А. – № 9. – С. 73–75.
8. Бодянский Е. В. Об одном адаптивном алгоритме обнаружения разладок в случайных последовательностях / Е. В. Бодянский, И. А. Руднева // Автолматика и телемеханика. – 1995. – № 10. – С. 101–106.
9. Бодянский Е. В. Алгоритм адаптивной идентификации динамических параметрически нестационарных объектов / Е. В. Бодянский, С. А. Воробьев, А. Штефан // Изв. РАН. Теория и системы управления. – 1999. – № 1. – С. 19–23.

© Бодянський Є. В., Тесленко Н. О.,
Дейнеко А. О., 2011

Стаття надійшла до редколегії 27.04.2011 р.