

УДК 681.513

Е.В. Горшков, В.В. Колодяжный, И.П. Плисс

АДАПТИВНЫЙ АЛГОРИТМ ОБУЧЕНИЯ НЕО-ФАЗЗИ НЕЙРОНА И НЕЙРО-ФАЗЗИ СЕТИ НА ОСНОВЕ ПОЛИНОМИАЛЬНЫХ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ

1. Введение

Последние годы для решения широкого класса задач обработки данных и управления в условиях структурной и параметрической неопределенности все чаще применяются искусственные нейронные сети и системы нечеткого вывода. На стыке этих двух направлений возникли нейро-фаззи сети, обладающие способностью к обучению, подобно нейросетям, и обеспечивающие интерпретируемость результатов, подобно нечетким системам. Именно это направление вычислительного интеллекта сейчас находится на стадии бурного развития. Достаточно близко к нейро-фаззи сетям приемыкает гораздо менее известная конструкция, называемая нео-фаззи нейрон [1–3], схема которого приведена на рис. 1.

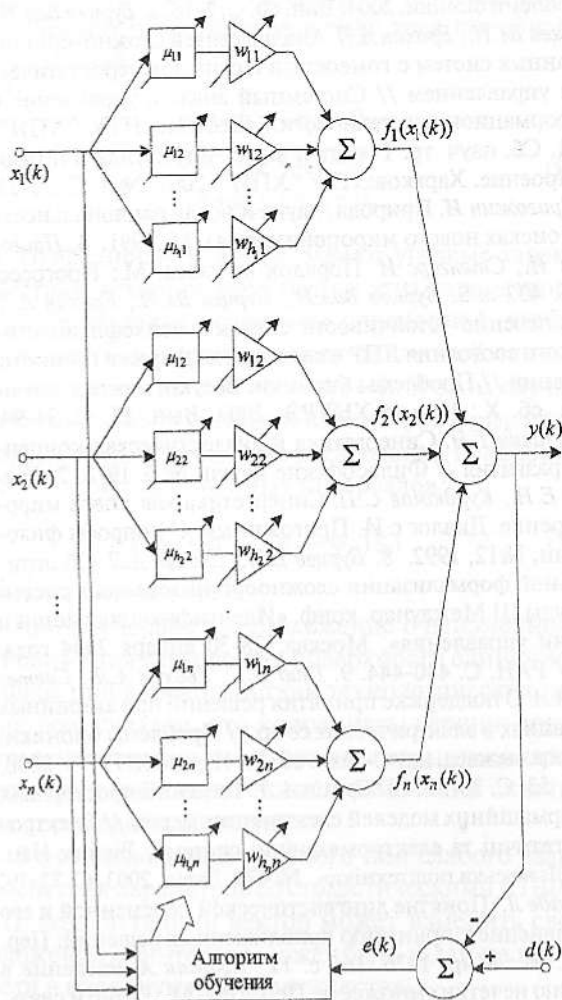


Рис. 1. Нео-фаззи нейрон

При подаче на вход нео-фаззи нейрона векторного сигнала-образа $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ (здесь $k = 1, 2, \dots$ – номер наблюдения в обучающей выборке или текущее дискретное время) на его выходе появляется скалярный сигнал

$$y(k) = \sum_{i=1}^n f_i(x_i(k)) = \sum_{i=1}^n \sum_{j=1}^{h_i} w_{ji}(k) \mu_{ji}(x_i(k)),$$

определяемый как значениями используемых функций принадлежности $\mu_{ji}(x_i(k))$, $j = 1, 2, \dots, h_i$; $i = 1, 2, \dots, n$, так и настраиваемыми синоптическими весами $w_{ji}(k)$.

В [4] были предложены обобщение нео-фаззи нейрона на случай векторного выходного сигнала $y(k)$ и градиентный алгоритм обучения синаптических весов $w_{ji}(k)$ с постоянным параметром шага, причем во всех случаях использовалась фиксированная треугольная функция принадлежности

$$\mu_{ji}^L(x_i(k)) = \begin{cases} 1 - d_{ji}^{-1} |c_{ji} - x_i(k)|, & \text{если } |c_{ji} - x_i(k)| \leq d_{ji}, \\ 0, & \text{в противном случае} \end{cases}$$

или, что то же самое,

$$\mu_{ji}^L(x_i(k)) = (1 - d_{ji}^{-1} |c_{ji} - x_i(k)|) \delta_{ji},$$

где c_{ji} , d_{ji} – параметры центра и ширины функции принадлежности μ_{ji} соответственно;

$$\delta_{ji} = \begin{cases} 1, & \text{если } |c_{ji} - x_i(k)| \leq d_{ji}, \\ 0, & \text{в противном случае.} \end{cases}$$

При этом в качестве критерия обучения использовалась стандартная квадратичная форма

$$E(k) = \frac{1}{2} (d(k) - y(k))^2 = \frac{1}{2} e^2(k) = \frac{1}{2} \left(d(k) - \sum_{i=1}^n \sum_{j=1}^{h_i} w_{ji}(k) \mu_{ji}(x_i(k)) \right)^2, \quad (1)$$

где $d(k)$ – обучающий сигнал.

В [5] для повышения скорости обучения был введен оптимальный по быстродействию одношаговый градиентный алгоритм, а для улучшения аппроксимирующих свойств было предложено использовать квадратичные функции принадлежности

$$\mu_{ji}^Q(x_i(k)) = (1 - d_{ji}^{-2} (c_{ji} - x_i(k))^2) \delta_{ji} \quad (2)$$

с настраиваемыми параметрами центра и ширины.

Проведенный анализ показал, что существенное увеличение скорости сходимости не сопровождалось значимым улучшением аппроксимирующих свойств, что, как выяснилось, связано с тем, что производные квадратичной функции принадлежности терпят разрыв в точке пересечения с осью абсцисс ($\frac{\partial \mu_{ji}^Q(x_i)}{\partial x_i} \rightarrow \pm d_{ji}^{-1}$ при $x_i \rightarrow c_{ji} \pm d_{ji}$). Именно это свойство побудило предпринять настоящее исследование.

2. Обучаемый нео-фаззи нейрон с полиномиальными функциями принадлежности

Введем в рассмотрение функции принадлежности в виде полинома четвертой степени

$$\mu_{ji}^F(x_i(k)) = (1 - d_{ji}^{-2}(c_{ji} - x_i(k))^2)^2 \delta_{ji}, \quad (3)$$

форма которых приведена на рис. 2.

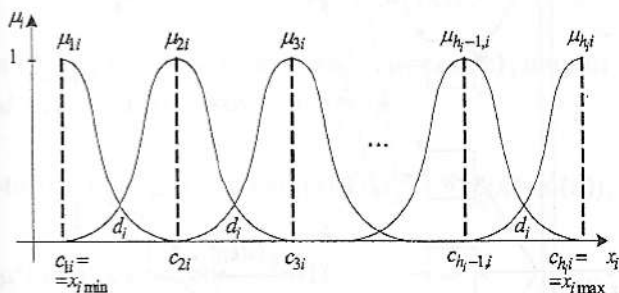


Рис. 2. Полиномиальные функции принадлежности

Как видно, это колоколообразные функции, производные которых не претерпевают разрыва на всей области определения.

Вводя в рассмотрение производные критерия (1) по настраиваемым параметрам с учетом (2) и (3)

$$\begin{cases} \frac{\partial E(k)}{\partial w_{ji}} = -e(k)\mu_{ji}^F(x_i(x)), \\ \frac{\partial E(k)}{\partial c_{ji}} = 4e(k)w_{ji}\mu_{ji}^Q(x_i(x))d_{ji}^{-2}(c_{ji} - x_i(k)), \\ \frac{\partial E(k)}{\partial d_{ji}^{-2}} = 2e(k)w_{ji}\mu_{ji}^Q(x_i(k))(c_{ji} - x_i(k))^2, \end{cases}$$

можно записать градиентный алгоритм обучения нео-фаззи нейрона с функциями принадлежности четвертой степени

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \eta^w(k)e(k)\mu_{ji}^F(x_i(k)), \\ c_{ji}(k+1) = c_{ji}(k) - \eta^c(k)e(k)w_{ji}(k)\mu_{ji}^Q(x_i(x)) \cdot d_{ji}^{-2}(k)(c_{ji}(k) - x_i(k)), \\ d_{ji}^{-2}(k+1) = d_{ji}^{-2}(k) - \eta^d(k)e(k)w_{ji}(k)\mu_{ji}^Q(x_i(k)) \cdot (c_{ji}(k) - x_i(k))^2, \end{cases} \quad (4)$$

где $\eta^w(k), \eta^c(k), \eta^d(k)$ — переменные параметры шага обучения, подлежащие определению.

Записывая далее $(h_i \times 1)$ -векторы параметров i -го нелинейного сигнала нео-фаззи нейрона

$$\mu_i^F(x(k)) = (\mu_{1i}^F(x_i(k)), \mu_{2i}^F(x_i(k)), \dots, \mu_{h_i}^F(x_i(k)))^T,$$

$$\mu_i^Q(x(k)) = (\mu_{1i}^Q(x_i(k)), \mu_{2i}^Q(x_i(k)), \dots, \mu_{h_i}^Q(x_i(k)))^T,$$

$$c_i(k) = (c_{1i}(k), c_{2i}(k), \dots, c_{h_i}(k))^T,$$

$$d_i^-(k) = (d_{1i}^{-2}(k), d_{2i}^{-2}(k), \dots, d_{h_i}^{-2}(k))^T,$$

$$w_i(k) = (w_{1i}(k), w_{2i}(k), \dots, w_{h_i}(k))^T$$

и $(h \times 1)$ -векторы параметров нео-фаззи нейрона

$$(h = \sum_{i=1}^n h_i)$$

$$\mu^F(x(k)) = (\mu_1^{FT}(x(k)), \mu_2^{FT}(x(k)), \dots, \mu_n^{FT}(x(k)))^T,$$

$$\mu^Q(x(k)) = (\mu_1^{QT}(x(k)), \mu_2^{QT}(x(k)), \dots, \mu_n^{QT}(x(k)))^T,$$

$$c(k) = (c_1^T(k), c_2^T(k), \dots, c_n^T(k))^T,$$

$$d^-(k) = (d_1^{-T}(k), d_2^{-T}(k), \dots, d_n^{-T}(k))^T,$$

$$w(k) = (w_1^T(k), w_2^T(k), \dots, w_n^T(k))^T,$$

$$X(k) = (x_1(k)E_1^T, x_2(k)E_2^T, \dots, x_n(k)E_n^T)^T$$

(здесь E_i — $(h_i \times 1)$ -вектор, состоящий из единиц), можно ввести векторную форму алгоритма (4) в виде

$$\begin{cases} w(k+1) = w(k) + \eta^w(k)e(k)\mu^F(x(k)), \\ c(k+1) = c(k) - \eta^c(k)e(k)w(k) \odot \mu^Q(x(k)) \odot d^-(k) \odot (c(k) - X(k)), \\ d^-(k+1) = d^-(k) - \eta^d(k)e(k)w(k) \odot \mu^Q(x(k)) \odot (c(k) - X(k)) \odot (c(k) - X(k)), \end{cases}$$

где \odot — символ скоттова произведения.

Для увеличения скорости сходимости процесса обучения следует перейти от градиентных процедур со скалярным параметром шага $\eta(k)$ к процедурам, использующим матричный коэффициент усиления, наиболее популярным из которых является алгоритм Левенберга-Марквардта.

С этой целью введем $(h \times 1)$ -векторы $\mu^c(x(k)) = 4w(k) \odot \mu^Q(x(k)) \odot d^-(k) \odot (c(k) - X(k))$ и $\mu^d(x(k)) = 2w(k) \odot \mu^Q(x(k)) \odot (c(k) - X(k)) \odot (c(k) - X(k))$, после чего можно записать

$$\begin{cases} w(k+1) = w(k) + e(k)(\mu^F(x(k))\mu^{FT}(x(k)) + \alpha^w I)^{-1} \cdot \mu^F(x(k)), \\ c(k+1) = c(k) - e(k)(\mu^c(x(k))\mu^{cT}(x(k)) + \alpha^c I)^{-1} \cdot \mu^c(x(k)), \\ d^-(k+1) = d^-(k) - e(k)(\mu^d(x(k))\mu^{dT}(x(k)) + \alpha^d I)^{-1} \cdot \mu^d(x(k)), \end{cases}$$

где $\alpha^w, \alpha^c, \alpha^d$ — неотрицательные регуляризирующие параметры; $I - (h \times h)$ - единичная матрица.

Используя далее технику преобразований, описанную в [6, 7], можно ввести модификацию алгоритма (5), обладающую как следящими, так и сглаживающими свойствами:

$$\left\{ \begin{aligned} w(k+1) &= w(k) + \frac{e(k)\mu^F(x(k))}{\alpha^w(k)}, \\ \alpha^w(k+1) &= \alpha\alpha^w(k) + \|\mu^F(x(k))\|^2, \\ c(k+1) &= c(k) - \frac{e(k)\mu^c(x(k))}{\alpha^c(k)}, \\ \alpha^c(k+1) &= \alpha\alpha^c(k) + \|\mu^c(x(k))\|^2, \\ d^-(k+1) &= d^-(k) - \frac{e(k)\mu^d(x(k))}{\alpha^d(k)}, \\ \alpha^d(k+1) &= \alpha\alpha^d(k) + \|\mu^d(x(k))\|^2, \\ 0 &\leq \alpha \leq 1. \end{aligned} \right. \quad (6)$$

Как видно, алгоритм (6), при $\alpha = 0$ совпадающий с процедурой Левенберга-Марквардта, не использует операцию обращения матриц, что резко упрощает его численную реализацию.

Можно отметить, что улучшение обобщающих свойств нео-фаззи нейрона в ряде случаев может быть обеспечено применением несимметричных функций принадлежности, у которых параметр ширины задается соотношениями

$$\left\{ \begin{aligned} d_{ji}^L(k) &= d_{ji}^L(k) + d_{ji}^R(k), \\ d_{ji}^L(k) &= c_{ji}(k) - c_{j-1,i}(k), \\ d_{ji}^R(k) &= c_{j+1,i}(k) - c_{ji}(k). \end{aligned} \right.$$

При этом автоматически отпадает необходимость в использовании третьего рекуррентного выражения в (6), а кроме того обеспечивается защита от возникновения нежелательных «мертвых зон» между соседними функциями принадлежности.

3. Нейро-фаззи сеть с полиномиальными функциями принадлежности

Достаточно близка по архитектуре к нео-фаззи нейрону нейро-фаззи сеть Ванга-Менделя [8], схема которой без выходного слоя нормализации приведена на рис. 3. Следует заметить, что если число настраиваемых синаптических весов нео-фаззи нейрона совпадает с размерностью входного пространства n , то нейро-фаззи сеть имеет h_j весов по числу функций принадлежности в antecedенте j -го входа и реализует отображение

$$y(k) = \sum_{j=1}^{h_j} w_j(k) f_j(x(k)) = \sum_{j=1}^{h_j} w_j(k) \prod_{i=1}^n \mu_{ji}(x_i(k)).$$

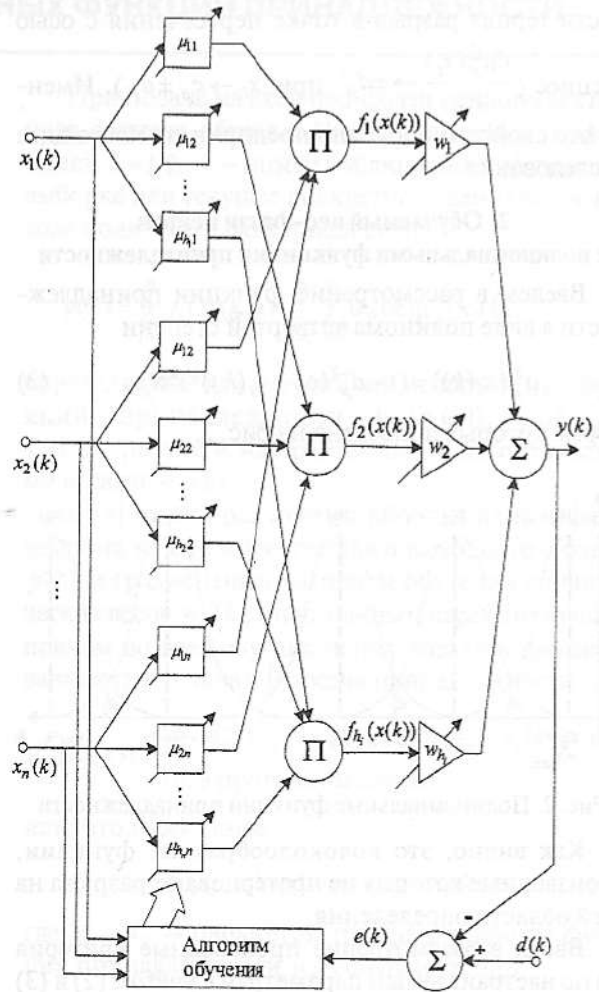


Рис. 3. Нейро-фаззи сеть

Вводя производные квадратичного критерия обучения по настраиваемым параметрам с учетом (3)

$$\left\{ \begin{aligned} \frac{\partial E(k)}{\partial w_j} &= -e(k) \prod_{i=1}^n \mu_{ji}^F(x_i(k)), \\ \frac{\partial E(k)}{\partial c_{ji}} &= 4e(k)w_j \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \mu_{ji}^Q(x_i(k)) d_{ji}^{-2} \cdot \\ &\quad \cdot (c_{ji} - x_i(k)), \\ \frac{\partial E(k)}{\partial d_{ji}^{-2}} &= 2e(k)w_j \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \mu_{ji}^Q(x_i(k)) \cdot \\ &\quad \cdot (c_{ji} - x_i(k))^2, \end{aligned} \right.$$

можно записать градиентный алгоритм обучения вида

4. Выводы

Предложен адаптивный алгоритм обучения нео-фаззи нейрона и нейро-фаззи сети с полиномиальными функциями активации, позволяющий настраивать в реальном времени не только синаптические веса, но и параметры введенных функций принадлежности четвертой степени, что обеспечивает высокое качество аппроксимации.

Алгоритм прост в вычислительном отношении, обладает высокой скоростью сходимости, а также следящими и сглаживающими свойствами.

Список литературы: 1. *Yamakawa T., Uchino E., Miki T., Kusanagi H.* A neo fuzzy neuron and its applications to system identification and prediction of the system behavior // Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA'92). Iizuka, Japan, 1992. P. 477-483. 2. *Uchino E., Yamakawa T.* Soft computing based signal prediction, restoration, and filtering // Da Ruan (Ed.): Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms. Kluwer Academic Publishers, Boston, 1997. P. 331-349. 3. *Miki T., Yamakawa T.* Analog implementation of neo-fuzzy neuron and its on-board learning // N.E. Mastorakis (Ed.): Computations Intelligence and Applications. WSES Press, Piraeus, 1999. P. 144-149. 4. *Landim R., Menezes B., Silva S.* A Neo-Fuzzy-Neuron with Real Time Training Allied to Flux Observer for an Induction Motor // Proc. V-th Brazilian Symp. On Neural Networks (SBRN'98). Belo Horizonte, Brazil, 1998. P. 67-72. 5. *Bodyanskiy Ye., Kokshenev I., Kolodyazhnyi V.* An adaptive learning algorithm for a neo fuzzy neuron // Proc. 3rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT 2003), Zittau, Germany, 10-12 September 2003. P. 375-379. 6. *Bodyanskiy Ye., Kolodyazhnyi V., Stephan A.* An adaptive learning algorithm for a neuro-fuzzy network // B. Reusch (Ed.): Computational Intelligence. Theory and Applications. Springer-Verlag, Berlin-Heidelberg-New York, 2001. P. 68-75. 7. *Otto P., Bodyanskiy Ye., Kolodyazhnyi V.* A new learning algorithm for a forecasting neuro-fuzzy network // Integrated Computer-Aided Engineering. 10. N4. Amsterdam: IOS Press, 2003. P. 399-409. 8. *Осовский С.* Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.

Поступила в редакцию 12.11.2004

$$\begin{cases} w_j(k+1) = w_j(k) + \eta^w(k)e(k) \prod_{i=1}^n \mu_{ji}^F(x_i(k)), \\ c_{ji}(k+1) = c_{ji}(k) - \eta^c(k)e(k)w_j(k) \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \cdot \mu_{ji}^Q(x_i(k))d_{ji}^{-2}(k)(c_{ji}(k) - x_i(k)), \\ d_{ji}^{-2}(k+1) = d_{ji}^{-2}(k) - \eta^d(k)e(k)w_j(k) \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \cdot \mu_{ji}^Q(x_i(k))(c_{ji} - x_i(k))^2. \end{cases} \quad (7)$$

Записывая далее $(h_i \times 1)$ -векторы

$$\mu^F(x(k)) = \left(\prod_{i=1}^n \mu_{1i}^F(x_i(k)), \prod_{i=1}^n \mu_{2i}^F(x_i(k)), \dots, \prod_{i=1}^n \mu_{h_i}^F(x_i(k)) \right)^T,$$

$$w(k) = (w_1(k), w_2(k), \dots, w_{h_i}(k))^T$$

и $(\sum_{i=1}^n h_i \times 1) = (h \times 1)$ -векторы центров $c(k)$, ширины $d^-(k)$, $\mu^c(x(k))$ с компонентами

$$4w_j(k) \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \mu_{ji}^Q(x_i(k))d_{ji}^{-2}(k)(c_{ji}(k) - x_i(k)),$$

$\mu^d(x(k))$ с компонентами

$$2w_j(k) \left(\prod_{p=1, p \neq j}^n \mu_{pi}^F(x_i(k)) \right) \mu_{ji}^Q(x_i(k))(c_{ji}(k) - x_i(k))^2,$$

можно перейти к векторной форме алгоритма обучения (7), полностью совпадающей с предложенной выше системой рекуррентных соотношений (6).