УДК 621.317.7
© *І. Стрепко*[1], *І. Рудик*[2], *Ю. Наконечний*[2], *2010*

АЛГОРИТМИ НАВЧАННЯ ТА ВИКОРИСТАННЯ ГІБРИДНИХ МЕРЕЖ

Розглянуто алгоритми навчання та використання гібридних мереж, а також показано їх переваги та недоліки у порівнянні з штучними нейронними мережами.

The article reviews the study of algorithms and using hybrid networks, and shows their advantages and disadvantages in comparison with artificial neural networks..

1. ВСТУП

Кожний різновид систем штучного інтелекту має свої особливості, наприклад, за можливостями навчання, узагальнення і вироблення висновків, що робить ці системи найпридатнішими для вирішення одного класу задач і менш придатними – для іншого.

Наприклад, нейронні мережі добре використовувати для задач розпізнавання образів, але вельми незручні для з'ясування питання, як вони таке розпізнавання здійснюють. Вони можуть автоматично набувати знання, але процес їх навчання часто відбувається достатньо поволі, а аналіз навченої мережі вельми складний. При цьому яку-небудь апріорну інформацію (знання експерта) для прискорення процесу її навчання в нейронну мережу ввести неможливо.

Системи з нечіткою логікою, навпаки, хороші для пояснення одержуваних з їх допомогою висновків, але вони не можуть автоматично набувати знання для використовування їх в механізмах висновків [1].

2. ПОСТАНОВКА ПРОБЛЕМИ

Розглянемо типовий підхід до побудови алгоритмів навчання і використання гібридних нейронних мереж.

Припустимо, що гібридною мережею має бути реалізоване (невідоме) відображення:

$$y^k = f(x^k) = f(x_1^k, x_2^k, ..., x_n^k), \quad k = 1, 2, ..., N,$$

при наявності навчальної множини $\{(x^1, y^1), ..., (x^N, y^N)\}$.

[1] Українська академія друкарства
[2] Національний університет "Львівська політехніка"

Для моделювання невідомого відображення f використаємо спрощений алгоритм нечіткого виводу із застосуванням наступної форми запису предикатних правил:

Π_i: якщо $x_1 \in A_{i1}$ і $x_2 \in A_{i2}$ і . . . , і $x_n \in A_{in}$, тоді $y = z_i$, $i = 1,2, \ldots, m$,

де A_{ij} − нечіткі числа трикутної форми , z_i − речові числа, визначаючи ступінь істинності i-го правила з допомогою операції множення:

$$\alpha_i = \prod_{j=1}^{n} A_{ij}(x_j^k)$$

(тут можна використовувати і інші подання для моделювання логічного оператора «І») і визначаючи вихід нечіткої системи дискретним аналогом центроїдного методу:

$$o^k = \frac{\sum_{i=1}^{m} \alpha_i z_i}{\sum_{i=1}^{m} \alpha_i}.$$

Введення функції помилки для k-го запропонованого зразка виду:

$$E_k = \frac{1}{2}(o^2 - y^k)^2$$

дозволяє надалі як в звичайних (стандартних) нейронних мережах використовувати градієнтний метод для підбору параметрів заданих предикатних правил[2]. Так, величини можна коректувати по співвідношенню:

$$z_i := z_i - \eta \frac{\partial E_k}{\partial z_i} = z_i - \eta (o^k - y^k) \frac{\alpha_i}{\alpha_1 + \alpha_2 + \ldots + \alpha_m}, \; i = 1,2, \ldots, m,$$

де η , як і раніше – константа яка характеризує швидкість навчання.

Більш детально алгоритм налагодження розглянемо на прикладі, який включає два правила:

Π_1: якщо $x \in A_1$ тоді $y = z_1$,

Π_2: якщо x є тоді A_2 тоді $y = z_2$,

При цьому припускається що нечіткі поняття A_1(«малий») і A_2 («великий») мають сигмоїдні функції приналежності:

$$A_1(x) = \frac{1}{1 + e^{b_1(x - \alpha_1)}}, \qquad A_2(x) = \frac{1}{1 + e^{b_2(x - \alpha_2)}},$$

які характеризуються параметрами $\alpha_1, \alpha_2, b_1, b_2$.

Ступені істинності правил визначаються в даному випадку співвідношенням:

$$\alpha_1 = A_1(x) = \frac{1}{1+e^{b_1(x-\alpha_1)}}, \qquad \alpha_2 = A_2(x) = \frac{1}{1+e^{b_2(x-\alpha_2)}},$$

а вихід системи – виразом:

$$o = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} = \frac{A_1(x) z_1 + A_2(x) z_2}{A_1(x) + A_2(x)}.$$

Припустимо, що ми маємо навчальну множину $\{(x^1, y^1), \ldots, (x^N, y^N)\}$, яка відображає невідому функцію f.

Необхідно: здійснити таке налагодження параметрів системи α_1, α_2, b_1, b_2, z_1, z_2, при якій забезпечується найкраща апроксимація даної функції.

Вирішення. В даному випадку функція помилки може бути записана в формі:

$$E_k = E_k(\alpha_1, b_1, \alpha_2, b_2, z_1, z_2) = \frac{1}{2}(o^k(\alpha_1, b_1, \alpha_2, b_2, z_1, z_2) - y^k)^2.$$

Використовуючи надалі такий самий підхід , що у алгоритму зворотного розповсюдження помилки запишемо розповсюдження помилки:

$$z_1 := z_1 - \eta \frac{\partial E_k}{\partial z_1} = z_1 - \eta(o^k - y^k)\frac{\alpha_1}{\alpha_1 + \alpha_2} =$$

$$= z_1 - \eta(o^k - y^k)\frac{A_1(x^k)}{A_1(x^k) + A_2(x^k)},$$

$$z_2 := z_2 - \eta \frac{\partial E_k}{\partial z_2} = z_2 - \eta(o^k - y^k)\frac{\alpha_1}{\alpha_1 + \alpha_2} =$$

$$= z_2 - \eta(o^k - y^k)\frac{A_2(x^k)}{A_1(x^k) + A_2(x^k)},$$

Аналогічним шляхом можна отримати розгорнуті вирази для корекції коефіцієнтів α_1, α_2, b_1, b_2 .

Вихідні співвідношення такі:

$$a_1 := a_1 - \eta \frac{\partial E_k}{\partial a_1}, \qquad a_2 := a_2 - \eta \frac{\partial E_k}{\partial a_2},$$

$$b_1 := b_1 - \eta \frac{\partial E_k}{\partial b_1}, \qquad b_2 := b_2 - \eta \frac{\partial E_k}{\partial b_2}$$

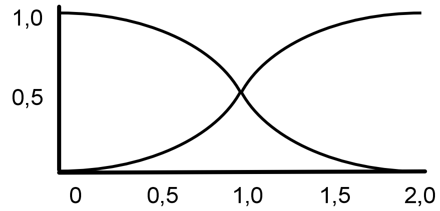

Рис. 1. Симетричні функції приналежності

Кінцеві вирази є досить великими , але можуть бути спрощені у випадку, якщо функції приналежності мають вигляд:

$$A_1(x) = \frac{1}{1+e^{-b(x-\alpha)}}, \qquad A_2(x) = \frac{1}{1+e^{b(x-\alpha)}},$$

Дані функції характеризуються двома параметрами (a і b) в певному сенсі є симетричними (рис.1.) і задовільняють рівнянню:

$$A_1(x) + A_2(x) = 1.$$

Зауважимо, що з попереднього і вище отриманих рівнянь випливає:

$$z_1 := z_1 - \eta \frac{\partial E_k}{\partial z_1} = z_1 - \eta(o^k - y^k)\frac{\alpha_1}{\alpha_1 + \alpha_2} = z_1 - \eta(o^k - y^k)A_1(x^k),$$

$$z_2 := z_2 - \eta \frac{\partial E_k}{\partial z_2} = z_2 - \eta(o^k - y^k)\frac{\alpha_2}{\alpha_1 + \alpha_2} = z_2 - \eta(o^k - y^k)A_2(x^k),$$

Наступні вирази такі:

$$a := a - \eta \frac{\partial E_k(a,b)}{\partial a},$$

де

$$\frac{\partial E_k(a,b)}{\partial a} = (o^k - y^k)\frac{\partial o^k}{\partial a} = (o^k - y^k)\frac{\partial}{\partial a}(z_1 A_1(x^k) + z_2 A_2(x^k)) =$$

$$= (o^k - y^k)\frac{\partial}{\partial a}(z_1 A_1(x^k) + z_2(1 - A_1(x^k))) =$$

$$= (o^k - y^k)(z_1 - z_2)\frac{\partial A_1(x^k)}{\partial a} = (o^k - y^k)(z_1 - z_2)b \times \frac{e^{b(x^k-\alpha)}}{(1+e^{b(x^k-\alpha)})^2} =$$

$$= (o^k - y^k)(z_1 - z_2)bA_1(x^k)(1 - A_1(x^k)) =$$

$$= (o^k - y^k)(z_1 - z_2)bA_1(x^k)A_1(x^k),$$

і $$b := b - \eta \frac{\partial E_k(a,b)}{\partial b},$$

де $$\frac{\partial E_k(a,b)}{\partial b} = (o^k - y^k)(z_1 - z_2)\frac{\partial}{\partial b}\frac{1}{e^{b(x^k-\alpha)}} =$$

$$= (o^k - y^k)(z_1 - z_2)(x^k - \alpha)A_1(x^k)(1 - A_1(x^k)) =$$

$$= (o^k - y^k)(z_1 - z_2)(x^k - \alpha)A_1(x^k)\,A_1(x^k)$$

Наведені вирази повністю ілюструють ідеї алгоритмів навчання і використання гібридної мережі.

Іншим прикладом може служити система, яка налічує наступну базу знань:

$П_1$: якщо x_1 є L_1 і x_2 є L_2 і x_3 є L_3 тоді y є H,

$П_2$: якщо x_1 є H_1 і x_2 є H_2 і x_3 є L_3 тоді y є M,

$П_3$: якщо x_1 є H_1 і x_2 є H_2 і x_3 є H_3 тоді y є S,

де $x_1,\ x_2,\ x_3$ − вхідні змінні, y − вихід системи, є $L_1,\ L_2,\ L_3,\ H_1,\ H_2,\ H_3,\ M,\ S$ − деякі нечіткі множини з функціями приналежності сигмоїдного типу:

$$L_j(t) = \frac{1}{1+e^{b_j(t-c_j)}},\quad H_j(t) = \frac{1}{1+e^{-b_j(t-c_j)}},\quad j = 1,2,3,$$

$$H(t) = \frac{1}{1+e^{-b_4(t-c_4+c_5)}},\quad M(t) = \frac{1}{1+e^{-b_4(t-c_4)}},$$

$$S(t) = \frac{1}{1+e^{b_4(t-c_4)}}.$$

Для визначення вихідної змінної використовується алгоритм виводу, тобто:

Підраховуються значення істинності для кожного правила:

$$\alpha_1 = L_1(\alpha_1) \wedge L_2(\alpha_2) \wedge L_3(\alpha_3),$$
$$\alpha_2 = H_1(\alpha_1) \wedge H_2(\alpha_2) \wedge L_3(\alpha_3),$$
$$\alpha_3 = H_1(\alpha_1) \wedge H_2(\alpha_2) \wedge H_3(\alpha_3),$$

де в даному випадку $\alpha_1,\ \alpha_2,\ \alpha_3$ − поточні значення виходів системи

для кожного правила визначаються окремі виходи:

$$z_1 = B^{-1}(\alpha_1) = c_4 + c_5 + \frac{1}{b_4}\ln\frac{1-\alpha_1}{\alpha_1},$$

$$z_2 = B^{-1}(\alpha_2) = c_4 + \frac{1}{b_4}\ln\frac{1-\alpha_2}{\alpha_2},$$

$$z_3 = B^{-1}(\alpha_3) = c_4 + \frac{1}{b_4} \ln \frac{1-\alpha_3}{\alpha_3},$$

знаходиться загальний вихід системи:

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3}{\alpha_1 + \alpha_2 + \alpha_3}.$$

Цей процес можна проілюструвати рис. 2.

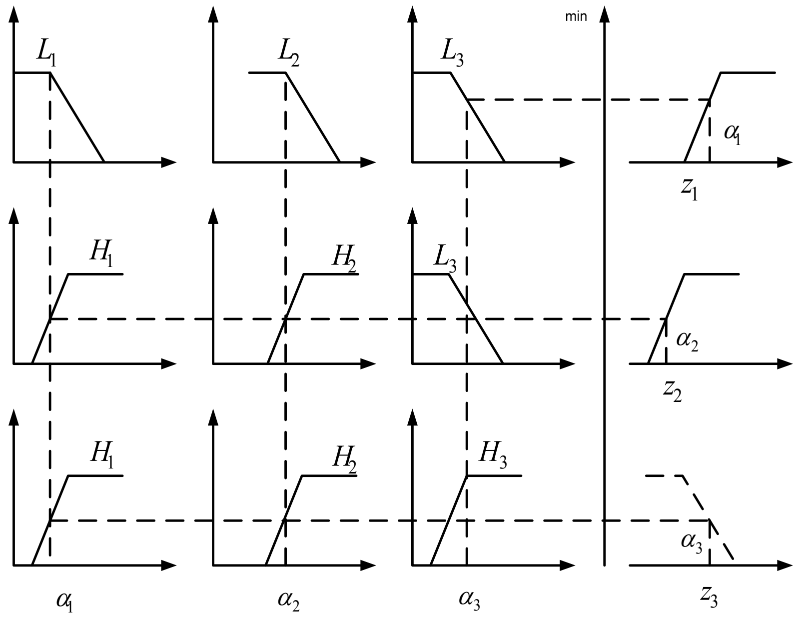

Рис. 2. Ілюстрація алгоритму виводу

Гібридна нейронна мережа, яка відображає наведений механізм виводу, подана на рис. 2. Зауважимо, що мережі з подібною архітектурою називають ANFIS (Adaptive Neuro-Fuzzy Inference System).

Дана мережа може бути описана таким чином.

Шар 1 (Layer 1). Виходи вузлів цього шару представляють собою значення функцій приналежності при конкретних (заданих) значеннях входів.

Шар 2 (Layer 2). Виходами нейронів цього шару є ступені істинності передумов кожного правила бази знань системи, які обчислюють за формулами:

$$\alpha_1 = L_1(\alpha_1) \wedge L_2(\alpha_2) \wedge L_3(\alpha_3),$$
$$\alpha_2 = H_1(\alpha_1) \wedge H_2(\alpha_2) \wedge L_3(\alpha_3),$$
$$\alpha_3 = H_1(\alpha_1) \wedge H_2(\alpha_2) \wedge H_3(\alpha_3),$$

Всі нейрони цього шару позначені буквою T, що означає, що вони можуть реалізовувати довільну *t*-норму для моделювання операції «І».

Шар 3 (Layer 3). Нейрони цього шару (позначені буквою N) обчислюють величини:

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \ \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \ \beta_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}$$

Шар 4 (Layer 4). Нейрони даного шару виконують операції:

$$\beta_1 z_1 = \beta_1 H^{-1}(\alpha_1), \quad \beta_2 z_2 = \beta_2 H^{-1}(\alpha_2), \quad \beta_3 z_3 = \beta_3 H^{-1}(\alpha_3), .$$

Шар 5 (layer 5). Єдиний нейрон цього шару обчислює вихід мережі:

$$z_0 = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3, .$$

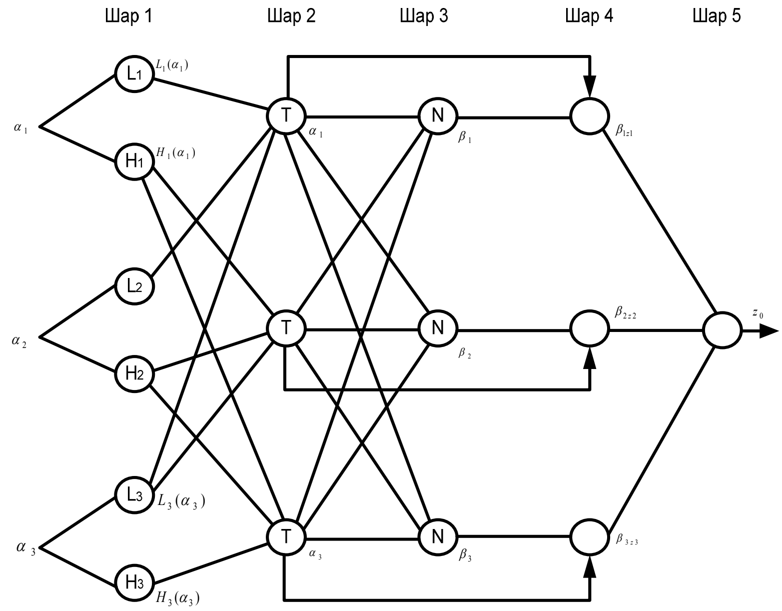

Рис. 3. Структура гібридної нейронної мережі (архітектура ANFIS)

Коригування параметрів системи тут виробляється у відповідності з раніше розглянутими підходом.

Так, наприклад, налагодження коефіцієнтів b_4, c_4 і c_5 - за формулами:

$$b_4 := b_4 - \eta \frac{\partial E_k}{\partial b_4} = b_4 - \frac{\eta}{b_4^2} \delta_k \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3},$$

$$c_4 := c_4 - \eta \frac{\partial E_k}{\partial c_4} = c_4 + \eta \delta_k \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} = c_4 + \eta\, \delta_k,$$

$$c_5 := c_5 - \eta \frac{\partial E_k}{\partial c_5} = c_5 + \eta \delta_k \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \text{ де } \delta_k = y^k - o^k.$$

Відповідні вирази можуть бути отримані і для інших коефіцієнтів.

3. ВИСНОВКИ

Системи з нечіткою логікою і штучні нейронні мережі еквівалентні одна одній, проте, на практиці у них є свої власні переваги і недоліки [3]. Дане міркування лягло в основу апарату гібридних мереж, в яких висновки робляться на основі апарату нечіткої логіки, але відповідні функції приналежності підлагоджуються використанням алгоритмів навчання нейронних мереж, наприклад, алгоритму зворотного розповсюдження помилки. Такі системи не тільки використовують апріорну інформацію, але можуть набувати нові знання і для користувача є логічно прозорими.

1. Роберт Коллан Основные концепции нейронных сетей — М., 2005. - С. 63-72. 2. Медведєв В., Потемкин В. Нейронные сети MATLAB6. С.П., 2006. - С. 73-87. 3. Круглов В.В., Борисов В.В. Искуственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия – Телеком, 2006. - С. 115-124.