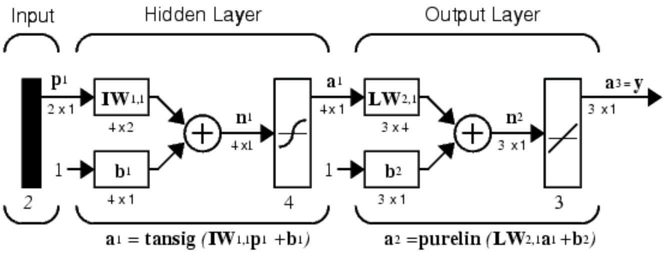
Algorytm wstecznej propagacji błędu

Ewa Adamus

ZUT
Wydział Informatyki
Instytut Sztucznej Inteligencji i Metod Matematycznych

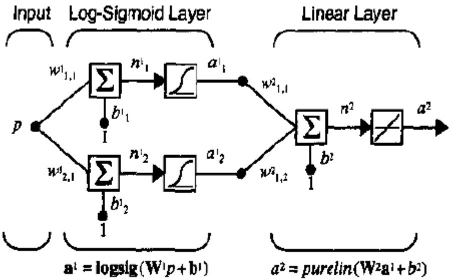
8 maja 2012

Sieć wielowarstwowa. Architektura



Rysunek: Architektura sieci służącej aproksymacji funkcji

Przykład sieci wielowarstwowej



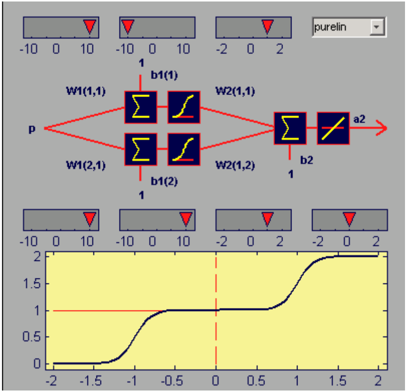
Równanie opisujące działanie sieci:

$$a^2 = purelin(W^2 * (logsig(W^1 * p + b^1)) + b^2) \qquad (1)$$

Funkcje przejścia neuronów:

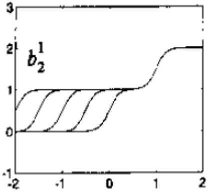
- w pierwszej warstwie typu: $f^1(n) = \frac{1}{1+e^{-n}}$,
- w drugiej warstwie: $f^2(n) = n$.

Wpływ parametrów sieci na jej odpowiedzi

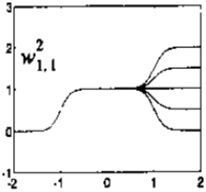


Rysunek: Przykład sieci oraz odpowiedzi dla przykładowych wartości parametrów

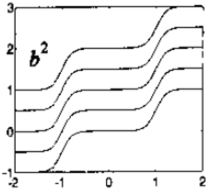
Wpływ parametrów sieci na jej odpowiedzi cd.



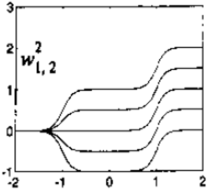
Wpływ b_2^1 (pierwsza warstwa, drugi neuron)



Wpływ w_{11}^2 (druga warstwa)



Wpływ b_2 (druga warstwa)



Wpływ $w_{12}^2 (drugawarstwa)$

Jak uczy się sieć wielowarstwowa?

Najbardziej popularny **algorytm wstecznej propagacji błędu** (zaproponowany w 1969 przez Brysona i Ho).
Algorytm posiada dwie fazy:

- próbka treningowa podawana na wejście i propagowana do wyjścia,
- obliczony błąd propagowany wstecz – parametry sieci modyfikowane.

Algorytm wstecznej propagacji

1. Inicjacja parametrów sieci.
2. Propagacja wartości wejściowych do wyjścia sieci.
3. Wartość błędu ($e = t - a$). Algorytm wstecznej propagacji koryguje parametry sieci tak, aby minimalizować powierzchnię funkcji błędu średniokwadratowego:

$$F(x) = E[e^2] = E[(t-a)^2]. \tag{2}$$

4. Korekcja parametrów sieci:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha * \frac{\delta F}{\delta w_{i,j}^m} \tag{3}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha * \frac{\delta F}{\delta b_i^m} \tag{4}$$

gdzie α jest współczynnikiem uczenia.

Korekcja parametrów sieci

Ponieważ $F(x)$ jest pośrednią funkcją wag oraz biasów w warstwie ukrytej, w celu ustalenia pochodnych przyjmijmy:

$$n_i^m = \sum w_{i,j}^m * a_j^{m-1} + b_i^m, \tag{5}$$

wejście do m-ej warstwy sieci, które jest bezpośrednio zależne od wag oraz biasów tej warstwy.
Tak więc pochodne możemy wyrazić w formie:

$$\frac{\delta F}{\delta w_{i,j}^m} = \frac{\delta F}{\delta n_i^m} \times \frac{\delta n_i^m}{\delta w_{i,j}^m} \tag{6}$$

$$\frac{\delta F}{\delta b_i^m} = \frac{\delta F}{\delta n_i^m} \times \frac{\delta n_i^m}{\delta b_i^m} \tag{7}$$

Korekcja parametrów sieci cd.

Ponieważ n jest bezpośrednio zależne od zmiennych w oraz b, więc z zależności 5 pochodne:

$$\frac{\delta n_i^m}{\delta w_{i,j}^m} = a_j^{m-1}, \ \frac{\delta n_i^m}{\delta b_i^m} = 1. \tag{8}$$

Tak więc zależności 6, 7 po przyjęciu, że:

$$s_i^m = \frac{\delta F}{\delta n_i^m}, \tag{9}$$

możemy przedstawić w postaci:

$$\frac{\delta F}{\delta w_{i,j}^m} = s_i^m * a_j^{m-1}, \tag{10}$$

$$\frac{\delta F}{\delta b_i^m} = s_i^m. \tag{11}$$

Korekcja parametrów sieci cd.

Ostatecznie, aktualizacja wag oraz biasów w formie macierzowej:

$$W^m(k+1) = W^m(k) - \alpha * s^m * (a^{m-1})^T, \tag{12}$$

$$b^m(k+1) = b^m(k) - \alpha * s^m. \tag{13}$$

Pozostało nam jeszcze do ustalenia wyrażenie s^m. Forma pochodnej w tym przypadku zależy od numeru bieżącej warstwy m. Rozpocznijmy od ostatniej warstwy, gdzie bezpośrednio ustalamy wartość błędu średniokwadratowego (wyr.2.):

$$s^M = \frac{\delta F}{\delta n^M} = -2 * F^M(n^M) * (t - a), \tag{14}$$

a następnie propagujemy do kolejnych warstw:
$m = M - 1, \ldots, 2, 1$ zgodnie z zależnością:

$$s^m = F^m(n^m) * (W^{m+1})^T * s^{m+1}. \tag{15}$$

Przykład – aproksymacja funkcji kwadratowej określona w przedziale $< -2, 2 >$:

$g(p) = (p-1)^2$ dla $-2 \leq p \leq 2$

- **Inicjujemy parametry sieci:**

$$
\begin{aligned}
W^1(0) &= \begin{bmatrix} 0.09 \\ -0.37 \end{bmatrix}, b^1(0) = \begin{bmatrix} -0.29 \\ -0.17 \end{bmatrix}, \\
W^2(0) &= \begin{bmatrix} -0.45 \\ 0.33 \end{bmatrix}, b^2(0) = [0.39]
\end{aligned}
$$

- **Obliczamy odpowiedź sieci dla**

$$a^0 = p = 1.$$

Przykład cd. – wyliczenie odpowiedzi sieci

Wektor wartości odpowiedzi pierwszej warstwy (ukrytej) a^1:

$$\begin{aligned} a^1 = f^1(W^1 a^0 + b^1) &= logsig\left(\begin{bmatrix} 0.09 \\ -0.37 \end{bmatrix} * [1] + \begin{bmatrix} -0.29 \\ -0.17 \end{bmatrix}\right) \\ &= logsig\left(\begin{bmatrix} -0.2 \\ -0.54 \end{bmatrix}\right) = \begin{bmatrix} \frac{1}{1+e^{0.2}} \\ \frac{1}{1+e^{0.54}} \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.37 \end{bmatrix} \end{aligned}$$

Przykład cd. – wyliczenie odpowiedzi sieci

Wyjście drugiej warstwy:

$$a^2 = f^2(W^2 a^1 + b^2) = purelin\left(\left[\begin{array}{cc} -0.45 & 0.33 \end{array}\right] * \left[\begin{array}{c} 0.45 \\ 0.37 \end{array}\right] + [0.39]\right) = [0.31]$$

Porównajmy odpowiedź sieci z wartością docelową.

$$e = t - a = (p-1)^2 - a^2 = (1-1)^2 - 0.31 = -0.31$$

Przykład cd. – propagacja wstecz wartości błędu, w celu modyfikacji parametrów sieci

Pochodne funkcji przejścia dla warstwy pierwszej (ukrytej):

$$f^{1}(n) = \frac{d}{dn}\left(\frac{1}{1+e^{-n}}\right) = \frac{e^{-n}}{(1+e^{-n})^2} = \left(1 - \frac{1}{1+e^{-n}}\right) * \frac{1}{1+e^{-n}} = (1-a^1)*(a^1)$$

oraz drugiej:

$$f^{2}(n) = \frac{d}{dn}(n) = 1.$$

Wartość – wynikająca z błędu naszej aproksymacji – propagowana wstecz:

$$s^2 = -2 * F^2(n^2) * (t-a) = -2 * [f^2(n^2)] * (-0.31) = -2 * 1 * (-0.31) = 0.62.$$

Dla pierwszej warstwy z zależności (15.):

$$\begin{aligned} s^1 &= F^1(n^1) * (W^2)^T * s^2 = \begin{bmatrix} (1-a_1^1)(a_1^1) & 0 \\ 0 & (1-a_2^1)(a_2^1) \end{bmatrix} * \begin{bmatrix} -0.45 \\ 0.33 \end{bmatrix} * [0.62] \\ &= \begin{bmatrix} (1-0.45)(0.45) & 0 \\ 0 & (1-0.37)(0.37) \end{bmatrix} * \begin{bmatrix} -0.45 \\ 0.33 \end{bmatrix} * [0.62] \\ &= \begin{bmatrix} 0.247 & 0 \\ 0 & 0.233 \end{bmatrix} * \begin{bmatrix} -0.279 \\ 0.205 \end{bmatrix} = \begin{bmatrix} -0.069 \\ 0.048 \end{bmatrix} \end{aligned}$$

Przykład cd. – propagacja wstecz wartości błędu, w celu modyfikacji parametrów sieci

Teraz możemy dokonać **modyfikacji parametrów sieci**. Dla uproszczenia przyjmiemy współczynnik uczenia $\alpha = 0.1$. Z zależności (12., 13.) mamy:

$$\begin{aligned}
W^2(1) &= W^2(0) - \alpha * s^2 * (a^1)^T = [-0.45\ 0.33] - 0.1 * [0.62] * [0.45\ 0.37] \\
&= [-0.4779\ 0.3071], \\
b^2(1) &= b^2(0) - \alpha * s^2 = [0.39] - 0.1 * [0.62] = 0.328, \\
W^1(1) &= W^1(0) - \alpha * s^1 * (a^0)^T = \begin{bmatrix} 0.09 \\ -0.37 \end{bmatrix} - 0.1 * \begin{bmatrix} -0.069 \\ 0.048 \end{bmatrix} * [1] \\
&= \begin{bmatrix} 0.0969 \\ -0.3748 \end{bmatrix} \\
b^1(1) &= b^1(0) - \alpha * s^1 = \begin{bmatrix} -0.29 \\ -0.17 \end{bmatrix} - 0.1 * \begin{bmatrix} -0.069 \\ 0.048 \end{bmatrix} = \begin{bmatrix} -0.2831 \\ -0.1748 \end{bmatrix}.
\end{aligned}$$

Zadanie do samodzielnego wykonania

Przedstawić wielowarstwową sieć perceptronową wraz z procesem strojenia, rozwiązującą problem XOR.