
METODY I ZASTOSOWANIA SZTUCZNEJ INTELIGENCJI

LABORATORIUM nr 02

Temat: **Seci jendokierunkowe wielowarstwowe.
Algorytm propagacji wstecznej.**

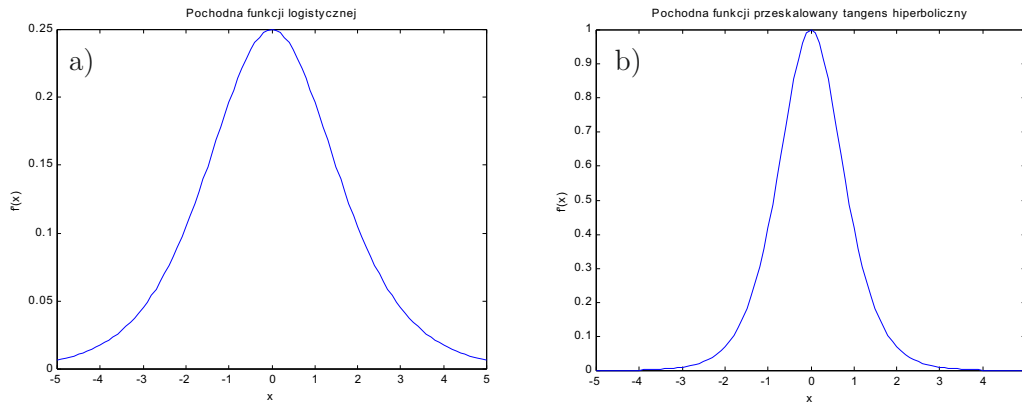
dr inż. Robert Tomkowski

Politechnika Koszalińska Wydział Mechaniczny
Katedra Mechaniki Precyzyjnej
pok. 118 bud. C, tel. 94 3478 251
robert.tomkowski@tu.koszalin.pl

Spis treści

| | |
|---|----------|
| 1. Wprowadzenie | 2 |
| 2. Algorytm propagacji wstecznej | 3 |
| 2.1. Algorytm największego spadku (Gradient Descent Algorithm – TRAINGD) | 4 |
| 2.2. Algorytm największego spadku z momentem (Gradient Descent Algorithm with Moment – TRAINGDM) | 4 |
| 2.3. Adaptacyjny dobór współczynnika uczenia (Gradient Descent with Adaptive LR Backpropagation – TRAINGDA, TRAINGDX) | 5 |
| 2.4. Algorytm Riedmilla i Brauna RPROP (Resilient Backpropagation - TRAINRP) | 5 |
| 3. Zadania | 6 |
| 3.1. Zadanie 1 | 6 |

Graficzne zestawienia pochodnych przedstawiono na rysunku 3.



Rys. 3 Graficzne przedstawienie pochodnej funkcji logistycznej

Kształt funkcji aktywacji ma mały wpływ na skuteczność sieci, ale kształt pochodnej funkcji aktywacji może znacząco wpłynąć na czas procesu uczenia. Mała pochodna funkcji aktywacji spowalnia uczenie sieci w przypadku zastosowania algorytmów propagacji wstecznej.

2. Algorytm propagacji wstecznej

Jednymi z najskuteczniejszych metod uczenia sieci jednokierunkowych wielowarstwowych są metody gradientowe. W metodach tych uaktualnienie wektora wag W i obciążenia b odbywa się zgodnie ze wzorem:

$$W(k+1) = W(k) + \Delta W \quad (5)$$

$$b(k+1) = b(k) + \Delta b \quad (6)$$

w którym

$$\Delta W = n \cdot p(W) \quad (7)$$

$$\Delta b = n \cdot p(b) \quad (8)$$

gdzie:

n - jest współczynnikiem uczenia;

$p(W), p(b)$ - kierunkiem w przestrzeni wielowymiarowej odpowiednio wag W i obciążenia b .

Uczenie sieci wielowarstwowej przy zastosowaniu metod gradientowych do wyznaczenia kierunku ($p(W), p(b)$), wymaga określenia gradientu względem wag (obciążenia) wszystkich warstw sieci. Jedynie w przypadku wag warstwy wyjściowej jest to zadanie określone w sposób bezpośredni. Warstwy pozostałe wymagają zastosowania specjalnej strategii postępowania, która w dziedzinie sieci neuronowych nosi nazwę algorytmu propagacji wstecznej (*backpropagation*).

Zgodnie z tym algorytmem w każdym cyklu uczącym wyróżnia się następujące etapy uczenia:

- Analiza sieci neuronowej o zwykłym kierunku przepływu sygnałów przy założeniu sygnałów wejściowych sieci równych elementom aktualnego wektora x . W wyniku analizy otrzymuje się wartości sygnałów wyjściowych neuronów warstw ukrytych oraz warstwy wyjściowej, a także odpowiednie pochodne $\frac{df(u_i^{(1)})}{du_i^{(1)}}$, $\frac{df(u_i^{(2)})}{du_i^{(2)}}$, ..., $\frac{df(u_i^{(n)})}{du_i^{(n)}}$ funkcji aktywacji w poszczególnych warstwach.

- Utworzenie sieci propagacji wstecznej przez odwrócenie kierunków przepływu sygnału, zastąpienie funkcji aktywacji przez ich pochodne, a także podanie do byłego wyjścia (obecnie wejścia) sieci wymuszenia w postaci odpowiedniej różnicy między wartością aktualną a żadaną. Dla tak utworzonej sieci należy obliczyć wartości odpowiednich różnic wstecznych.
- Adaptacja wag i obciążenia (uczenie sieci) odbywa się na podstawie wyników uzyskanych w powyższych podpunktach dla sieci zwykłej i sieci o propagacji wstecznej według odpowiednich wzorów.
- Omówiony proces należy powtórzyć dla wszystkich wzorców uczących, kontynuując go do chwili spełnienia warunków zatrzymania algorytmu.

2.1. Algorytm największego spadku (Gradient Descent Algorithm – TRAINGD)

Najprostszą metodą wyboru kierunku minimalizacji $p(W), b(W)$, jest wybór zgodny z kierunkiem ujemnego gradientu:

$$\Delta W = -\nabla E(W) \quad (9)$$

$$\Delta b = -\nabla E(b) \quad (10)$$

W Matlabie znajduje się siedem parametrów połączonych z tą metodą uczenia:

- Epochs** - po osiągnięciu liczby epok (*epochs*) określonej przez ten warunek uczenie zostanie przerwane;
- Show** - po ilu iteracjach zostanie wyświetlony status procesu uczenia;
- Goal** - po spadku wartości błędu poniżej wartości określonej przez parametr *goal* uczenie zostanie przerwane;
- Time** - jeżeli proces uczenia będzie trwał dłużej niż wartość *time* określona w sekundach, uczenie zostanie przerwane;
- Min_grad** - jeżeli wartość gradientu osiągnie wartość określoną przez *min_grad* uczenie zostanie przerwane;
- Max_fail** - jeżeli błąd danych weryfikujących wzrośnie powyżej *max_fail* uczenie zostanie przerwane;
- Lr** - współczynnik uczenia.

Wadą tej metody uczenia jest brak postępów minimalizacji w okolicy punktu optymalnego, gdzie gradient przyjmuje bardzo małe wartości. Zaletą zaś jest niezwykła prostota tej metody, małe wymagania co do pamięci i stosunkowo mała złożoność obliczeniowa.

2.2. Algorytm największego spadku z momentem (Gradient Descent Algorithm with Moment – TRAINGDM)

Zastosowanie momentu przynosi poprawę efektywności algorytmu największego spadku. W przypadku tej metody aktualizacja wagi i obciążenia sieci odbywa się zgodnie ze wzorem:

$$\Delta W = m \cdot W(k-1) + (1-m) \cdot n \cdot (-\nabla E(W)) \quad (11)$$

$$\Delta b = m \cdot b(k-1) + (1-m) \cdot n \cdot (-\nabla E(b)) \quad (12)$$

gdzie:

m - jest współczynnikiem momentu $[0, 1]$;

n - jest współczynnikiem uczenia.

Jeżeli wartość współczynnika momentu wynosi zero, to wzór ten odpowiada standardowej metodzie największego spadku. Im większa jest wartość współczynnika, tym składnik wynikający z momentu ma większy wpływ na dobór wag. Jego wpływ wzrasta w sposób istotny na płaskich odcinkach funkcji celu oraz w pobliżu minimum lokalnego, gdzie wartość gradientu jest bliska zero.

Przy małej wartości gradientu czynnik momentu może stać się dominujący we wzorze i spowodować przyrost wag ΔW i obciążenia Δb odpowiadający wzrostowi wartości funkcji celu, umożliwiającemu opuszczenie określonego minimum lokalnego. Czynnika momentu nie może przy tym całkowicie zdominować procesu uczenia, gdyż prowadziłoby to do niestabilności algorytmu. Zwykle kontroluje się wartość funkcji celu E procesie uczenia, dopuszczając do jej wzrostu jedynie w ograniczonym zakresie. Za wzrost ten odpowiada skojarzony z algorytmem uczenia współczynnik *max_perf_inc* (standardowo 1,04). W przypadku jeżeli w kolejnych iteracjach k oraz $k + 1$ jest spełniona relacja $E(k + 1) < 1,04 \cdot E(k)$, to krok jest akceptowany i następuje uaktualnienie wartości wag i obciążenia, w przeciwnym wypadku wartość momentu m równa jest zero.

W takim przypadku składnik gradientowy odzyskuje dominację nad czynnikiem momentu i proces przebiega zgodnie z kierunkiem minimalizacji wyznaczonym przez wektor gradientu.

2.3. Adaptacyjny dobór współczynnika uczenia (Gradient Descent with Adaptive LR Backpropagation – TRAINGDA, TRAINGDX)

W przypadku standardowego algorytmu największego spadku wartość współczynnika uczenia jest stała podczas całego procesu optymalizacyjnego. Inną, bardziej skuteczną metodą jest zastosowanie adaptacyjnego zmian współczynnika uczenia. W metodzie tej współczynnik uczenia utrzymuje się na możliwie najwyższym poziomie, gwarantującym stabilność uczenia. Zmiana wartości współczynnika uczenia jest bezpośrednio zależna od błędu odpowiedzi sieci. Jeżeli błąd uczenia sieci w kolejnym kroku jest większy niż błąd w kroku poprzednim (przy czym zakłada się, że niewielki wzrost, nie przekraczający 4% jest dopuszczalny), to współczynnik uczenia zostanie zmniejszony (o wartość *lr_inc* - standardowo 0,7), w przeciwnym wypadku współczynnik uczenia jest zwiększany (o wartość *lr_dec* - standardowo 1,05).

W Matlabie zaimplementowano dwie funkcje wykorzystujące adaptacyjny dobór współczynnika uczenia: *traingda* - algorytm największego spadku i *traingdx* - algorytm największego spadku z momentem.

2.4. Algorytm Riedmillera i Brauna RPROP (Resilient Backpropagation - TRAINRP)

W algorytmie RPROP przy zmianie wag uwzględnia się jedynie znak składowej gradientu, pomijając jej wartość:

$$\Delta W_{ij}(k) = -n_{ij}^{(k)} \cdot \text{sgn} \left(\frac{\partial E(W(k))}{\partial W_{ij}} \right) \quad (13)$$

Współczynnik uczenia jest dobierany w każdym cyklu uczącym dla każdej wagi W_{ij} indywidualnie na podstawie zmian wartości gradientu:

$$n_{ij}^k = \begin{cases} \min(a \cdot n_{ij}^{k-1}, n_{max}) & \text{dla } S_{ij}(k)S_{ij}(k-1) > 0 \\ \max(b \cdot n_{ij}^{k-1}, n_{min}) & \text{dla } S_{ij}(k)S_{ij}(k-1) < 0 \\ n_{ij}^{(k-1)} & \text{w pozostałych przypadkach} \end{cases} \quad (14)$$

przy czym $S_{ij}(k) = \frac{\partial E(W(k))}{\partial W_{ij}}$, a i b są stałymi: $a = 1,2$, $b = 0,5$, natomiast n_{min} i n_{max} oznaczają odpowiednio minimalną i maksymalną wartość współczynnika uczenia, równo w algorytmie RPROP odpowiednio 10^{-6} oraz 50, a funkcja *sgn* oznacza znak argumentu (ang. *sign*).

Algorytm RPROP, pomijający informację o wartości gradientu, umożliwia znaczne przyspieszenie procesu uczenia w tych obszarach, gdzie nachylenie funkcji celu jest niewielkie. Strategia doboru wag

zakłada ciągły wzrost współczynnika uczenia, jeżeli w dwóch krokach znak gradientu jest taki sam, natomiast jego redukcję, gdy ten znak jest różny.

3. Zadania

3.1. Zadanie 1

W procesie wglębnego obwodowego szlifowania powierzchni walcowych wykonano 20 elementów przy zróżnicowanych wartościach prędkości obwodowej przedmiotu obrabianego $V_p(0 - 2\frac{m}{s})$ i dosuwu a ($0 - 1 mm$). Obrobione elementy sprawdzono pod kątem występowania niedokładności dyskwalifikujących wyrób. Za krytyczne niedokładności przyjęto: uszkodzenia cieplne oraz nadmierną falistość powierzchni obrobionej. Wyniki kontroli jakości zestawiono w tabeli 1.

Tab. 1 Zestawienie danych eksperymentalnych

| numer elementu obrob. | prędkość obwodowa przedm. $V_p [m/s]$ | dosuw $a [mm]$ | obróbka | numer elementu obrob. | prędkość obwodowa przedm. $V_p [m/s]$ | dosuw $a [mm]$ | obróbka |
|-----------------------|---------------------------------------|----------------|---------------|-----------------------|---------------------------------------|----------------|---------------|
| 1 | 1.99 | 0.17 | nieprawidłowa | 11 | 1.77 | 0.11 | nieprawidłowa |
| 2 | 1.00 | 0.71 | nieprawidłowa | 12 | 1.49 | 0.21 | prawidłowa |
| 3 | 1.30 | 0.10 | prawidłowa | 13 | 1.25 | 0.83 | nieprawidłowa |
| 4 | 0.79 | 0.41 | nieprawidłowa | 14 | 1.41 | 0.44 | prawidłowa |
| 5 | 0.58 | 0.92 | nieprawidłowa | 15 | 1.78 | 0.43 | prawidłowa |
| 6 | 1.30 | 0.61 | prawidłowa | 16 | 1.92 | 0.60 | prawidłowa |
| 7 | 1.88 | 0.98 | nieprawidłowa | 17 | 1.64 | 0.73 | prawidłowa |
| 8 | 1.05 | 0.28 | prawidłowa | 18 | 0.07 | 0.14 | nieprawidłowa |
| 9 | 0.61 | 0.15 | prawidłowa | 19 | 0.24 | 0.30 | nieprawidłowa |
| 10 | 1.16 | 0.50 | prawidłowa | 20 | 1.65 | 0.91 | nieprawidłowa |

Na podstawie danych z powyższej tabeli utworzyć sieć wielowarstwową umożliwiającą klasyfikację wybranych parametrów obróbki do dwóch kategorii: (i) kategorii zapewniającej poprawność obróbki z uwagi na brak występowania uszkodzeń cieplnych przedmiotu i nadmiernej falistości powierzchni obrobionej, (ii) kategorii w wyniku której obrobione przedmioty nie spełniają powyższych kryteriów jakościowych.

Zakres poprawnych parametrów obróbki przedstawić w formie graficznej.

Podpowiedź. Wykorzystać funkcje **newff**, **train**, **sim**.

Czy projektowany zbiór uczący musi zawierać wszystkie dane eksperymentalne (jeżeli nie, to jakimi kryteriami należy kierować się przy ich wyborze)? Czy możliwa jest klasyfikacja parametrów obróbki do więcej niż dwóch kategorii (od czego zależna jest liczba możliwych kategorii)? Czy możliwa jest poprawna klasyfikacja parametrów obróbki przy użyciu perceptronu bądź też sieci jednowarstwowej (uzasadnij odpowiedź)? Jaka jest minimalna struktura sieci zapewniająca poprawne rozwiązanie?