Metody detekcji uszkodzeń oparte na wiedzy — Sieci neuronowe (c.d.)

Dariusz Uciński

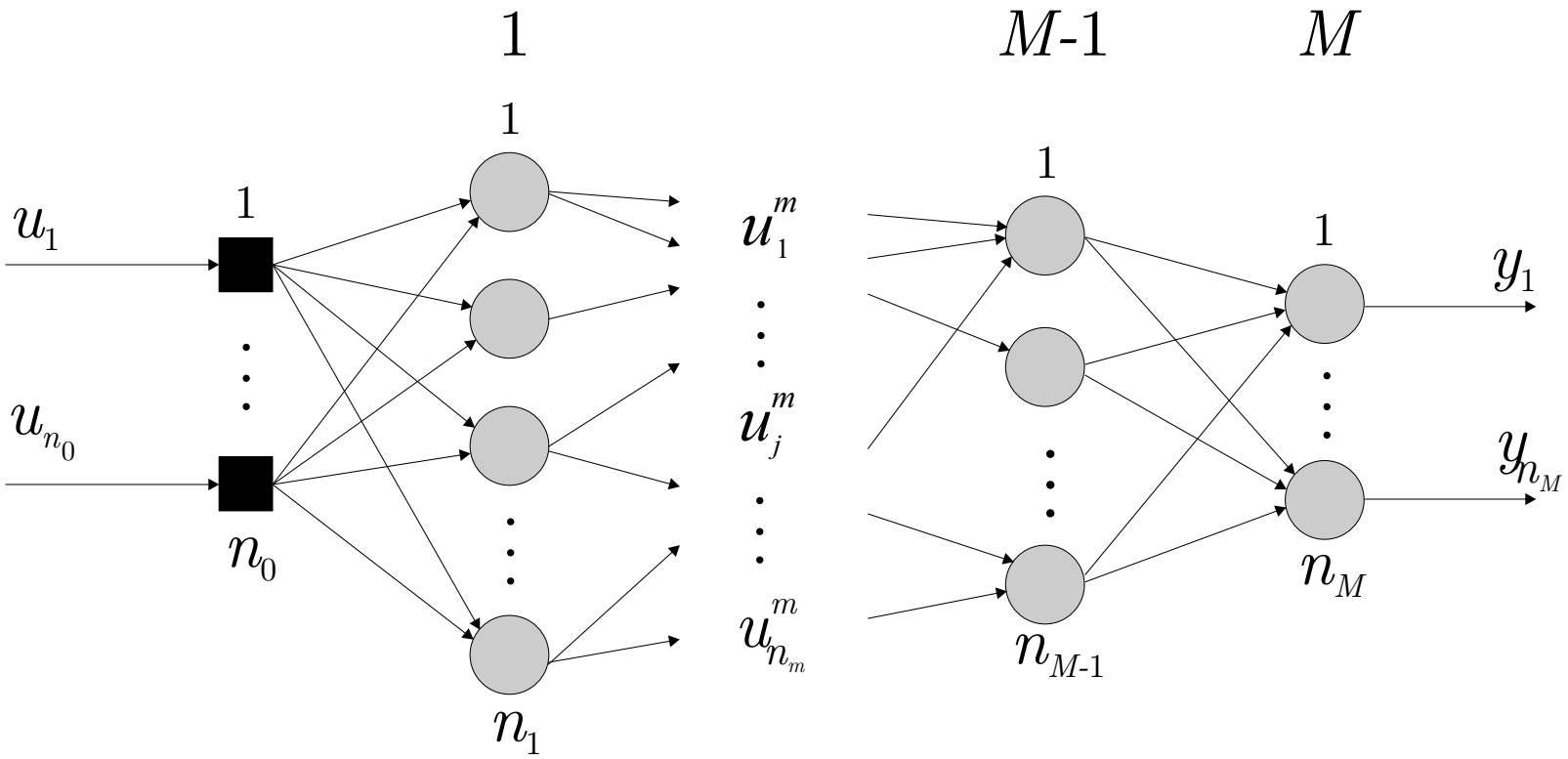
Instytut Sterowania i Systemów Informatycznych
Uniwersytet Zielonogórski

Wykład 8

Wprowadzenie

- Ograniczenia perceptronów prostych nie posiadają sieci z tzw. warstwami ukrytymi
- Sieć z jedną warstwą ukrytą może reprezentować dowolną funkcję boolowską (łącznie z problemem XOR)
- Sieci wielowarstwowe — ogromne możliwości aproksymujące
- *Problem:* jak uczyć taką sieć?
- *Rozwiązanie:* algorytm wstecznej propagacji błędu

Struktura sieci



Struktura sieci

Oznaczenia:

M — liczba warstw ($m = 0, \ldots, M$)

n_m — liczba neuronów w m-tej warstwie

w_{ji}^m — waga połączenia pomiędzy j-tym elementem m-tej warstwie i i-tym elementem $m-1$-tej warstwy

b_j^m — wartość progowa j-tego elementu m-tej warstwy

u_j — sygnał wejściowy na j-tym wejściu

y_j — sygnał wyjściowy na j-tym wyjściu

Struktura sieci

- Każdą warstwę można przedstawić za pomocą nieliniowwgo operatora aktywacji neuronów

$$N_m\{\mathbf{u}_m\} = F\{\mathbf{W}^m\mathbf{u}_m\} \qquad m = 1, \ldots, M$$

gdzie:

$\mathbf{u}_m$ — wektor sygnałów wejściowych dla m-tej warstwy

N_m — operator przetwarzania neuronowego m-tej warstwy

F — operator aktywacji neuronów

$\mathbf{W}^m$ — macierz współczynników wagowych pomiędzy warstwami m i $m-1$

Struktura sieci

- Odwzorowanie całej sieci wielowarstwowej:

$$\mathbf{y} = N_{wyj} N_m N_{m-1} \dots N_1 \{\mathbf{u}\}$$

$$\mathbf{y} = N\{\mathbf{u}\} = F_{wyj}\{\mathbf{W}^{wyj} F\{\mathbf{W}^m F\{\mathbf{W}^{m-1} \dots F\{\mathbf{W}^1 \mathbf{u}\} \dots\}\}\}$$

gdzie:

- $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ — wektor wyjściowy
- $\mathbf{W}^{wyj}$ — macierz wag pomiędzy warstwą wyjściową i m-tą
- F^{wyj} — operator aktywacji neuronów wyjściowych
- N^{wyj} — operator przetwarzania neuronowego warstwy wyjściowej

Uniwersalny aproksymator

- Sieć wielowarstwową można traktować jako układ do aproksymacji funkcji nieliniowych wielu zmiennych $\mathbf{y} = f(\mathbf{u})$
- Twierdzenie (Cybenko)

> Użycie co najmniej dwóch warstw ukrytych umożliwia osiągnięcie dowolnej dokładności aproksymacji dowolnej funkcji, przy dostatecznie dużej liczbie neuronów w warstwach

- Twierdzenie (Cybenko; Hornik *i in.*)

> Dowolna ciągła nieliniowa funkcja może być zaproksymowana z żądaną dokładnością za pomocą sieci neuronowej składającej się z tylko jednej warstwy ukrytej zawierającej dostatecznie dużą liczbę elementów

Uniwersalny aproksymator

- Użyteczność twierdzeń o uniwersalnym aproksymatorze zależy od tego ile wymaganych jest jednostek ukrytych
- W wielu przypadkach liczba jednostek ukrytych może wzrastać wykładniczo wraz ze wzrostem liczby wejść sieci
- Rozwiązanie: algorytmy doboru optymalnej struktury sieci

Wsteczna propagacja błędów

- Rozważmy M-warstwową sieć elementów o jednakowych ciągłych i różniczkowalnych funkcjach aktywacji $f(\cdot)$
- Niech $u_i^{m\mu}$ — wyjście i-tego elementu m-tej warstwy przy prezentacji μ-tego wzorca $(m = 0, \ldots, M; i = 0, 1, \ldots, n_m; \mu = 1, 2, \ldots, P)$
- Aktywność j-tego elementu m-tej warstwy

$$u_j^{m\mu} = f(\varphi_j^{m\mu}) = f\left(\sum_{i=0}^{n_{m-1}} w_{ji}^m u_i^{(m-1)\mu}\right)$$

Wsteczna propagacja błędów

- Miara błędu

$$J = \sum_{\mu=1}^{P} J_\mu, \qquad J_\mu = \frac{1}{2} \sum_{j=1}^{n_M} \left(y_j^{z\mu} - y_j^{\mu} \right)^2$$

- Korekta wag $\mathbf{w}_{ji}^m$

$$\Delta^\mu w_{ji}^m = -\eta \frac{\partial J_\mu}{\partial w_{ji}^m} = -\eta \frac{\partial J_\mu}{\partial \varphi_j^{m\mu}} \frac{\varphi_j^{m\mu}}{\partial w_{ji}^m}$$

- Druga z pochodnych cząstkowych

$$\frac{\partial \varphi_j^{m\mu}}{\partial w_{ji}^m} = u_i^{m\mu}$$

Wsteczna propagacja błędów

- Pierwsza z pochodnych cząstkowych

$$\delta_j^{m\mu} = -\frac{\partial J_\mu}{\partial \varphi_j^{m\mu}} = -\frac{\partial J_\mu}{\partial u_j^{m\mu}} \frac{\partial u_j^{m\mu}}{\partial \varphi_j^{m\mu}} = -\frac{\partial J_\mu}{\partial u_j^{m\mu}} f'\left(\varphi_j^{m\mu}\right)$$

  - Dla warstwy wyjściowej

$$\frac{\partial J_\mu}{\partial u_j^{m\mu}} = -\left(y_j^{z\mu} - y_j^{\mu}\right) = -e_j^{\mu}$$

  - Dla warstw ukrytych wartość $u_j^{m\mu}$ jest jednym ze składników sum ważonych liczonych we wszystkich elementach warstwy $m+1$, stąd

$$\frac{\partial J_\mu}{\partial u_j^{m\mu}} = \sum_{l=1}^{n_{m+1}} \frac{\partial J_\mu}{\partial \varphi_l^{(m+1)\mu}} \frac{\partial \varphi_l^{(m+1)\mu}}{\partial u_j^{m\mu}}$$

$$= \sum_{l=1}^{n_{m+1}} -\delta_l^{(m+1)\mu} \frac{\partial}{\partial u_j^{m\mu}} \sum_{i=1}^{n_m} w_{li}^{(m+1)} u_i^{m\mu} = \sum_{l=1}^{n_{m+1}} -\delta_l^{(m+1)\mu} w_{li}^{(m+1}$$

Wsteczna propagacja błędów

- Każdemu elementowi przetwarzającemu możemy przypisać błąd
  - dla warstwy wyjściowej

$$\delta_j^{M\mu} = f'\left(\varphi_j^{M\mu}\right) e_j^{\mu}$$

  - dla warstw ukrytych

$$\delta_j^{m\mu} = f'\left(\varphi_j^{m\mu}\right) \sum_{l=1}^{n_{m+1}} \delta_l^{m+1}{}\mu w_{lj}^{(m+1)}$$

- Uogólniona reguła korekcji wag

$$\Delta^{\mu} w_{ji}^{m} = \eta \delta_j^{m\mu} u_i^{(m-1)\mu}$$

- Korekta po zaprezentowaniu całego zbioru wzorców

$$\Delta w_{ji}^{m} = \eta \sum_{\mu=1}^{P} \delta_j^{m\mu} u_i^{(m-1)\mu}$$

Algorytm wstecznej propagacji

Krok 1. Wybierz początkowe wartości wag jako małe liczby losowe

Krok 2. Wybierz wzorzec $\mathbf{u}^{\mu}$ i podaj go na wejście sieci

Krok 3. Wyznacz wartości wyjść $u_j^{m\mu}$ każdego elementu dla kolejnych warstw przetwarzania neuronowego

$$u_j^{m\mu} = f\left(\sum_{i=0}^{n_{m-1}} w_{ji}^{m} u_i^{(m-1)\mu}\right)$$

Krok 4. Oblicz wartości błędów e_j^{μ} i sumy kwadratów błędów J_{μ}

Krok 5. Oblicz wartości błędów dla warstwy wyjściowej $\delta_j^{M\mu}$

$$\delta_j^{M\mu} = f'\left(\varphi_j^{M\mu}\right) e_j^{\mu}$$

Algorytm wstecznej propagacji

Krok 6. Dokonaj wstecznej propagacji błędu wyjściowego $\delta_j^{M\mu}$ do poszczególnych elementów warstw ukrytych wyznaczając $\delta_j^{m\mu}$

$$\delta_j^{m\mu} = f'\left(\varphi_j^{m\mu}\right) \sum_{l=1}^{n_{m+1}} \delta_l^{m+1}\mu w_{lj}^{(m+1)}$$

Krok 7. Dokonaj aktualizacji wag kolejno pomiędzy warstwą wyjściową i ukrytą, a następnie pomiędzy warstwami ukrytymi, przesuwając się w kierunku warstwy wejściowej

$$\Delta^{\mu} w_{ji}^{m} = \eta \delta_j^{m\mu} u_i^{(m-1)\mu}$$

Krok 8. Jeżeli warunek stopu jest spełniony to STOP, w innym przypadku idź do Kroku 2 i dokonaj prezentacji kolejnego wzorca

Algorytm wstecznej propagacji — podsumowanie

- Wagi są używane zarówno do obliczania odpowiedzi siec jak również do wyznaczania błędów
- Można zauważyć analogię występującą pomiędzy wzorami:

$$u_j^m = f\left(\sum_{i=0}^{n_{m-1}} w_{ji}^m u_i^{(m-1)}\right), \quad \delta_j^m = f'\left(\varphi_j^m\right) \sum_{l=1}^{n_{m+1}} \delta_l^{(m+1)} w_{lj}^{m+1}$$

- Reguła modyfikacji jest „lokalna”, tzn. aby policzyć zmianę wagi danego połączenia należy znać wielkości dostępne tylko na obu końcach tego połączenia
- Złożoność obliczeniowa algorytmu nie jest duża. Jeżeli liczba wszystkich połączeń jest równa n, to obliczenie funkcji kosztu wymaga wykonania liczby operacji rzędu n i obliczenie wszystkich pochodnych wymaga liczby operacji rzędu n

Momentum

- Przyspieszenie zbieżności procesu uczenia przy jednoczesnym zachowaniu stabilności
- Metoda wyciszająca oscylacje wokół punktu optymalnego
- Każdej wadze nadaje się pewną bezwładność

$$\Delta \mathbf{w}(k) = -\eta \nabla E(k) + \alpha \Delta \mathbf{w}(k-1)$$

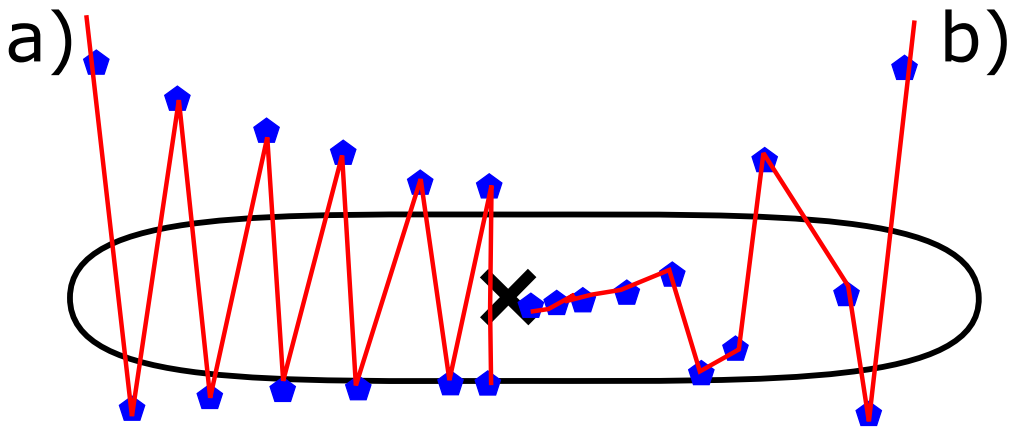
gdzie: α — współczynnik momentum o wartości $\alpha \in (0, 1)$

- Człon momentum działa jak filtr dolnoprzepustowy — eliminacja gwałtownych zmian wartości funkcji celu

Momentum

Przykład tłumienia oscylacji

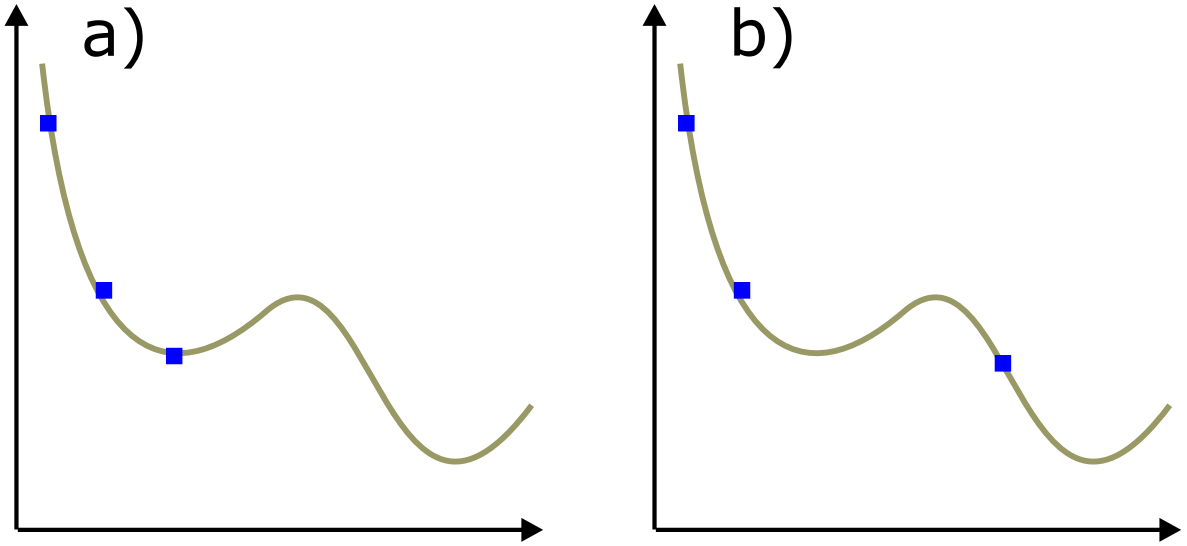
- przypadek (a) — proces uczenia dla $\eta = 0.0476$ bez momentum dla 12 kroków
- przypadek (b) — proces uczenia dla $\eta = 0.0476$ z członem momentum $\alpha = 0.5$ dla 12 kroków



Momentum

Przykład pokonania minimum lokalnego

- przypadek (a) — proces uczenia bez momentum
- przypadek (b) — proces uczenia z członem momentum

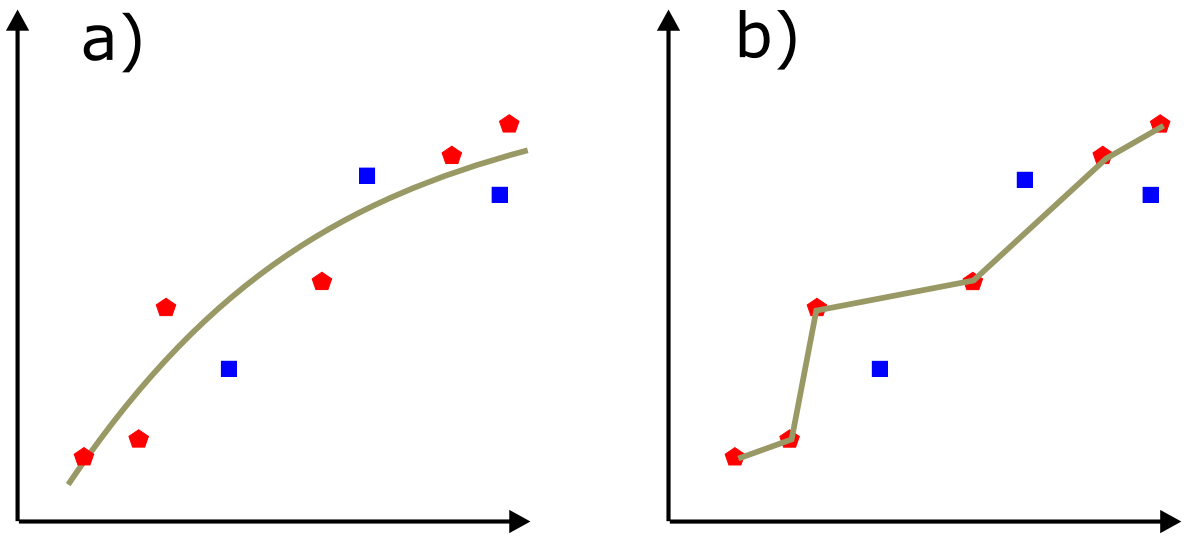


Parametry adaptacyjne

- Problem z wyborem odpowiednich wartości kroku uczenia
- Wartości η dobre w początkowej fazie uczenia mogą okazać się nieprawidłowe później
- Zaproponowano wiele rozwiązań
  - $$\eta(k) = \begin{cases} +a & \text{dla } \Delta E < 0 \\ -b\eta & \text{dla } \Delta E > 0 \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$
  - różne współczynniki uczenia dla każdej wagi
    $$\eta \simeq \frac{1}{\text{obciążalność węzła}}$$
  - zastosowanie po jednym parametrze η dla każdego wzorca

Uogólnianie

- W trakcie procesu uczenia można wykorzystać dowolną liczbę wzorców uczących
- W praktyce, do uczenia używa się zazwyczaj stosunkowo niewiele wzorców, a przy pomocy pozostałych sprawdza się czy nauczona sieć realizuje prawidłowo odwzorowanie
- Zbiór wzorców dzieli się na dwa podzbiory: uczący U i testujący T
- Zdolność do uogólnień polega na tym, że przy danym zbiorze uczącym sieć znajduje pewne wspólne cechy wektorów uczących

Uogólnianie

- Podanie wektora wejściowego, który nie był prezentowany w trakcie uczenia powoduje zaklasyfikowanie go do jednej z określonych klas wektorów, a wyjście sieci powinno być zbliżone do wyjścia odpowiadającego najbliższemu wzorcowi w przestrzeni wejść
- Porównanie dobrych właściwości uogólniania (a) i złych (b)