

Sieci neuronowe dwuwarstwowe

Napisz zestaw trzech funkcji MATLABowych do symulacji sieci dwuwarstwowych:

- **init2.m** – funkcja inicjująca sieć,

czyli tworząca jej macierz wag i wypełniająca ją wartościami losowymi z zakresu $-0.1 \div 0.1$
tu już każdy neuron będzie miał dodatkowe wejście zerowego – bias

przyjmuje: liczbę wejść sieci – S
liczbę neuronów w pierwszej warstwie – K1
liczbę neuronów w drugiej warstwie (liczbę wyjść) – K2
zwraca: macierz wag warstwy pierwszej – W1
macierz wag warstwy drugiej – W2

- **dzialaj2.m** – funkcja symulująca działanie sieci,

czyli dla sieci o danych macierzach wag W1 i W2
oblicza wektory wyjść kolejnych warstw Y1 oraz Y2 dla podanego na wejście wektora X

neurony wszystkich warstw mają sigmoidalne funkcje aktywacji: $y = f(u) = \frac{1}{1 + e^{-\beta u}}$

gdzie u jest pobudzeniem neuronu (sumą ważoną)

przyjmuje: macierz wag warstwy pierwszej – W1
macierz wag warstwy drugiej – W2
wektor wejść do sieci – X
zwraca: wektor wyjść sieci (drugiej warstwy) – Y2
wektor wejść pierwszej warstwy – Y1
tak naprawdę, to ta funkcja powinna zwracać tylko Y2 (wyjście sieci),
ale przekonacie się pisząc funkcję ucz2,
że ze względów technicznych musi również zwracać Y1

- **ucz2.m** – funkcja ucząca sieć na zadanym ciągu uczącym

ciąg uczący podany jest w postaci dwóch macierzy – wejść P i żądanych wyjść T
jednemu przykładowi uczącemu odpowiada para odpowiednich kolumn z macierzy P i T
czyli: macierz P – tyle kolumn, ile przykładów; tyle wierszy, ile wejść do sieci
macierz T – tyle kolumn, ile przykładów; tyle wierszy, ile wyjść sieci

funkcja przyjmuje: macierze wag sieci (przed uczeniem) – W1 i W2
macierz przykładów uczących (wejścia) – P
macierz przykładów uczących (żądane wyjścia) – T
liczba epok uczących – n
funkcja zwraca: macierze sieci (nauczonej) – W1 i W2

schemat funkcji (wystarczy zapisać w MATLABie, nie kombinować ☺):

```
% losuj przykład
% podaj go na wejścia i oblicz wyjścia
% oblicz błędy na wyjściach warstwy 2
% oblicz domniemane błędy wyjść warstwy 1 (backpropagation)
% oblicz poprawki wag warstw 1 i 2
% dodaj poprawki wag do wag warstw 1 i 2
% i to wszystko n razy
```

- ważna kolejność!
- ważna kolejność!
- dowolna kolejność
- dowolna kolejność

uwaga: obliczanie domniemanych błędów warstwy pierwszej
można zrobić mniej lub bardziej elegancko (sprytnie i matlabowo)

Sieć uczymy na przykładzie problemu XOR:

$$P = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix};$$

$$T = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$$

Ten ciąg uczący też jest skonstruowany jest minimalistycznie – dla czterech punktów, jakie sieć ma się nauczyć klasyfikować, ciąg składa się z czterech przykładów – dokładnie tych punktów.

To już nie jest proste zadanie – ono jest liniowo nieseparowalne. Należy sprawdzić, jak je rozwiązuje sieć jednowarstwowa (z poprzednich zajęć), jak sieć optymalna oraz sieci większe od optymalnej.

Co można zrobić jako dodatek (z ciekawości i na lepszą ocenę):

- ciąg uczący złożony z danych zaszumionych (takie chmurki wokół tych punktów wzorowych)
- wykres błędu średniokwadratowego obu warstw sieci w poszczególnych epokach (podczas uczenia sieci)
- adaptacyjny współczynnik uczenia: $\eta[i+1] = \begin{cases} \rho_d \cdot \eta[i] & \text{gdy } d[i] > k_w \cdot d[i-1] \\ \rho_i \cdot \eta[i] & \text{else} \end{cases}$

gdzie: $\eta[i]$ – współczynnik uczenia w epoce i -tej

$d[i]$ – błąd średniokwadratowy sieci w epoce i -tej

$k_w = 1.04$ – dopuszczalna wyżka błędu w kolejnych epokach

$\rho_d = 0.7$ – współczynnik redukcji współczynnika uczenia

$\rho_i = 1.05$ – współczynnik zwiększenia współczynnika uczenia

poobserwować prędkość uczenia sieci z takim adaptacyjnym współczynnikiem i porównać do zwykłego (np. dla dwóch sieci startujących od tych samych wag i uczonych na takiej samej sekwencji przykładów uczących)

- dla już nauczonej sieci można zrobić wykres zależności funkcji celu (błąd średniokwadratowy na danym ciągu uczącym) od którejś z wag – zaznaczyć na nim wagę osiągniętą przez sieć w procesie uczenia, można też zrobić wykres 3D zależności funkcji celu od dwóch którejkolwiek wag, a już pełnym bajerem jest pokazanie sekwencji takich zależności podczas procesu uczenia – czyli pokazanie jak funkcja celu osiąga minimum
- można też zrobić wykres granic decyzyjnych poszczególnych neuronów warstwy pierwszej i drugiej – dla już nauczonej sieci – przebieg (uczenie się) tych granic podczas uczenia