

Sieci neuronowe jednowarstwowe

Napisz zestaw trzech funkcji MATLABowych do symulacji sieci jednowarstwowych:

- init1.m – funkcja inicjująca sieć,**

czyli tworząca jej macierz wag i wypełniająca ją wartościami losowymi z zakresu $-0.1 \div 0.1$ na razie w sieciach jednowarstwowych nie wykorzystujemy wejścia zerowego – bias'u

przyjmuje: liczbę wejść sieci – S
liczbę neuronów w warstwie (liczbę wyjść) – K
zwraca: macierz wag warstwy – W

- dzialaj1.m – funkcja symulująca działanie sieci,**

czyli dla sieci o danej macierzy wag W
oblicza wektor wyjść Y dla podanego na wejście wektora X

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

gdzie u jest pobudzeniem neuronu (sumą ważoną)

przyjmuje: macierz wag warstwy – W
wektor wejść do sieci – X
zwraca: wektor wyjść sieci – Y

- ucz1.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: macierz wag sieci (przed uczeniem) – W
macierz przykładów uczących (wejścia) – P
macierz przykładów uczących (żądane wyjścia) – T
liczba epok uczących – n
funkcja zwraca: macierz sieci (nauczonej) – W

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

```
% losuj przykład
% podaj go na wejścia i oblicz wyjścia
% oblicz błędy na wyjściach
% oblicz poprawki wag
% dodaj poprawki wag do wag
% i to wszystko n razy
```

Sieć uczymy na przykładzie rozpoznawania zwierząt z Tadeusiewicza (str. 69):

```
% wejścia sieci (wiersze):
P = [ 4      2      -1      ; % ile ma nóg
      0.01   -1      3.5    ; % czy żyje w wodzie
      0.01    2      0.01   ; % czy umie latać
      -1     2.5     -2      ; % czy ma pióra
      -1.5    2      1.5    ] % czy jest jajorodny
% żądane wyjścia sieci (wiersze):
% (trzy kategorie -> trzy wyjścia
% trzy przykłady -> trzy kolumny)
T = [ 1  0  0 ; % ssak
      0  1  0 ; % ptak
      0  0  1 ] % ryba
```

Ciąg uczący skonstruowany jest minimalistycznie (dla tak prostego zadania, to wystarczy)
– dla trzech kategorii zwierząt, jakie się ma się nauczyć rozpoznawać,
ciąg składa się z trzech przykładów typowych przedstawicieli poszczególnych klas.

Jest to proste zadanie – liniowo separowalne, którego sięć bez problemu potrafi się nauczyć.

W ramach testu można na wejście sieci podać całą macierz P,
sieć powinna odpowiedzieć „prawie” macierzą T.

W ten sposób można testować tylko tę sieć z tego ćwiczenia – zbieg okoliczności, korzystny układ macierzy.
Normalnie, to aby przetestować sieć, trzeba by na jej wejścia podawać kolejne przykłady – czyli tu kolejne kolumny macierzy P,
i z otrzymanych wektorów wyjść utworzyć sobie macierz Y – porównać ją z macierzą T – powinny być prawie identyczne.
Oczywiście, w poważnych zastosowaniach, sieć testuje się na ciągach: sprawdzającym i testującym,
o czym była mowa na zajęciach.

Z ciekawości można sprawdzić, co sieci odpowiedzą dla takich danych:

- lis: x = [4 ; % 4 nogi
-1 ; % nie wodny
-1 ; % nie lata
-0.9; % nie ma piór, no chyba że akurat niesie gęś
-1] % nie jajorodny
- lis: x = [8 ; % 4 nogi, ale jest to tak ważne, że trzeba podać dwa razy
-6 ; % nie cierpi wody!
-3 ; % absolutnie nie lata, choć czasem by chciał...
-5 ; % nie ma piór, tylko rude futro!
-9] % nie składa jajek i wyprasza sobie takie podejrzenia
- foka, nietoperz, dziobak – ssaki
- struś – ptak
- ryba latająca
- wąż, żółw – ???

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

- inny przykład, np. XOR (żeby pokazać, że się nie da)
- wykres błędu średniokwadratowego sieci w poszczególnych epokach (podczas uczenia sieci)
- 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 którychkolwiek wag,
a już pełnym bajerem jest pokazanie sekwencji takich zależności podczas procesu uczenia
– czyli pokazanie jak funkcja celu osiąga minimum