

Zadanie 3 - Perceptron Rosenblatta (klasyfikacja)

Marcin Pietrzykowski

mpietrzykowski@wi.zut.edu.pl

wersja 1.0

1 Cel

Celem zadania jest zapoznanie się z zadaniem klasyfikacji i algorytmem Perceptronu Rosenblatta. Program jest do napisania w środowisku Matlab. Osoby chętne chcące napisać ww. program w innym języku odsyłam do punktu 6.

2 Zadanie klasyfikacji

Zadanie klasyfikacji polega na przydzielaniu obserwacji do z góry założonych klas, bazując na ich atrybutach (cechach). Jako przykład klasyfikacji rozpatrzmy proste zadanie mające na celu klasyfikację pacjenta na podstawie temperatury ciała (x_1) i procentowej zawartości liczby limfocytów w osoczu krwi (x_2) do jednej z dwóch klas *chory* ($y = -1$) lub *zdrowy* ($y = 1$)

Tablica 1: Przykładowa tabela z danymi

temperatura	% udział limfocytów	chory/zdrowy
x_1	x_2	y
36.6	0.35	zdrowy
37.8	0.52	chory
36.3	0.2	zdrowy
36.2	0.6	chory
...

Tablica 1 prezentuje przykładowy zbiór danych, na którym uczony jest algorytm. Po nauczeniu algorytm powinien być w stanie sklasyfikować pacjenta na *chorego/zdrowego* jedynie na podstawie podanych atrybutów tj. *temperatury* i *procentowego udziału limfocytów w osoczu*. Oczywiście jest to jedynie przykładowy zbiór danych, rzeczywiste zbiory danych zawierają o wiele większą liczbę atrybutów, a klasy z reguły nie są separowalne liniowo. Ponadto zbiór danych w bardziej zaawansowanych algorytmach z reguły jest dzielony na dane testujące, uczące i walidujące.

3 Algorytm Perceptronu Rosenblatta

Przytoczony tutaj algorytm perceptronu Rosenblatta ma na celu klasyfikację danych separowalnych liniowo tzn. takich gdzie pomiędzy dwoma klasami jesteśmy w stanie przeprowadzić linię prostą (płaszczyznę lub hiperpłaszczyznę) oddzielającą obie klasy. Przy założeniu, że po jednej stronie granicy znajdują się tylko instancje jednej klasy, a po drugiej stronie instancje drugiej klasy. Schemat Perceptronu przedstawia Rysunek 1. Sposób działania w szczególności został omówiony na zajęciach.

Zbiór danych wejściowych używany do uczenia perceptronu matematycznie możemy zapisać następująco:

$$\{(x_i, y_i)\}_{i=1..I}$$
$$x_i = (1, x_{i1}, x_{i2}, \dots, x_{in}) \in \mathbb{R}^n, \quad y_i = \{-1, 1\}$$

gdzie przez I oznaczamy liczbę danych wejściowych (instancji, próbek), a przez n liczbę atrybutów (cech). Uproszczony algorytm uczenia dla danych separowalnych liniowo ma postać:

1. $W(0) = (0, 0, \dots, 0)$
2. Przeiteruj się przez wszystkie próbki i wypełnij zbiór A wszystkimi próbkami danych, które zostały źle sklasyfikowane, czyli: $A = \{x_i, y_i \neq f(w(k) * x_i)\}$,
3. Jeżeli zbiór A jest pusty to zakończ.
4. Wylosuj próbkę ze zbioru A i na jej podstawie skoryguj wektor wag według wzoru $w(k+1) = w(k) + \eta * y_j * x_j$
5. $k = k + 1$
6. Wróć do punktu 2

Gdzie:

- $W = (w_0, w_1, \dots, w_n)$ to wagi perceptronu
- $w(k)$ to wagi w kroku k
- $\eta \in (0, 1]$ to współczynnik uczenia

4 Program

Na zajęciach należy napisać program klasyfikujący zbiór danych przedstawiony na Rysunku 2 przy pomocy perceptronu Rosenblatta i rysujący liniową granicę separacji.

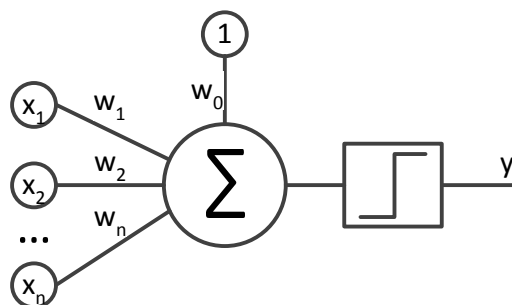
4.1 Dane wejściowe

Na początku skryptu należy zamieścić informacje takie jak:

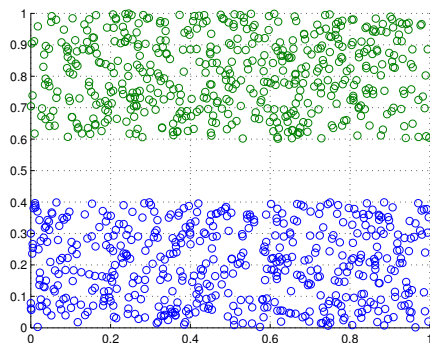
```

1  % liczba danych
2  I = 1000;
3  % liczba atrybutow
4  n = 2;
5  % wspolczynnik uczenia
6  eta = 0.5;
7  %Wagi perceptronu
8  W = zeros (1,n+1);
9  %licznik iteracji
10 k = 0;

```



Rysunek 1: Schemat Perceptronu.



Rysunek 2: Zbiór danych do klasyfikacji.

Następnie należy wygenerować zbiór danych wejściowych aby odwzorowywał zbiór danych z Rysunku 2. Zbiór powinien zawierać 1000 próbek po 500 na klasę. Rozmieszczenie próbek w klasach powinno być losowe. Dane powinny mieć postać macierzy X o wymiarach 1000×3 , wiersz i powinien być postaci $x_i = (1, x_{i1}, x_{i2})$ czyli każda próbka danych powinna posiadać 1 na wejściu, współrzędną x_1 i x_2 . Powinien zostać utworzony także wektor Y o rozmiarze 1000×1 definiujący do której klasy należy wybrana próbka danych. Można założyć że punkty zielone powinny posiadać wartość $y = -1$, a niebieskie wartość $y = 1$. Co można zapisać matematycznie w następujący sposób

$$y_i = \begin{cases} -1 & \text{jeżeli } x_{i2} > 0.6 \\ 1 & \text{jeżeli } x_{i2} < 0.4 \end{cases} \quad (1)$$

Następnie należy wyrysować zbiory na ekranie.

4.2 Główna pętla programu

W głównej pętli programu powinien znaleźć się algorytm podany w Punkcie 3. W celu obliczenia odpowiedzi bloku sumowania perceptronu wystarczy użyć następującego polecenia:

```
1 s = W * x';
```

4.3 Wyświetlenie granicy separacji

Następnie należy dodać fragment kodu odpowiedzialny za wyświetlenie odnalezionej granicy separacji

```
1 x1 = 0:0.05:1;
2 x2 = -(W(1) + W(2) * x1) / W(3) ;
3 hold on
4 plot (x1, x2, 'r');
```

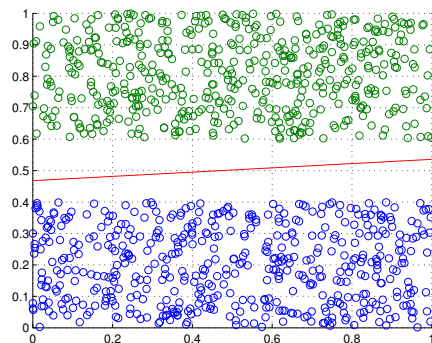
W wyniku działania algorytmu powinniśmy uzyskać zbiór wraz z granicą separacji o wyglądzie jak na Rysunku 3.

Wyżej utworzony skrypt należy zmodyfikować aby rysował granicę separacji w każdej iteracji uczenia algorytmu. A następnie przebadac jak zmienia się granica separacji w zależności od wartości parametru η .

Uwaga. Ze względu na fragment algorytmu, w którym do korekcji wag używa się losowo wybranej próbki za każdym razem granica separacji będzie posiadała trochę inne współrzędne.

5 Do domu

W domu należy wykonać implementację perceptronu w wersji nieliniowej. Treść zadania należy pobrać tutaj.



Rysunek 3: Zbiór danych wraz z granicą separacji.

6 Wersją do osób niełubiących Matlba

Osoby chętne mogą napisać ww. program w języku innym niż Matlab jednakże niezbędne będzie wyświetlenie wykresów 2D i 3D. Oprócz zastosowania zewnętrznej biblioteki natywnej dla danego języka rysującej wykresy istnieje możliwość skorzystania z Matlab'a jedynie w celu rysowania wykresów bezpośrednio w kodzie programu. Przykład jak można to wykonać w aplikacji konsolowej napisanej w języku C# jest do pobrania tutaj. Zaprezentowane rozwiązanie wymaga instancji Matlab'a zainstalowanej na maszynie, na której uruchamiany jest program.