

1 Perceptron binarny

Przedmiot: Sieci neuronowe i ich zastosowania

1.1 Perceptron binarny – przypomnienie

Najprostszą siecią jednokierunkową jest perceptron prosty, zbudowany jedynie z warstwy wejściowej i wyjściowej. Ponieważ nie istnieją połączenia pomiędzy elementami warstwy wyjściowej, każdy z tych elementów może być traktowany jako osobna sieć o $m + 1$ wejściach i jednym wyjściu.

Perceptrony strojone są na podstawie zbioru przykładów o określonym zachowaniu. To celowe zachowanie może być opisane jako zbiór par wektorów wejściowych z wyjściowymi: $p_1 t_1, p_2 t_2, p_3 t_3, \dots, p_q t_q$ gdzie p - wejście sieci, t - docelowe wyjście. Docelowy wektor t musi zawierać odpowiedzi typu 0 lub 1 — dla unipolarnej funkcji przejścia, bądź 0, -1 (funkcja bipolarna). Cały proces uczenia polega na takiej regulacji współczynników wagowych oraz biasów aby zminimalizować wartość błędu $e = t - a$, gdzie a jest faktyczną odpowiedzią sieci.

Algorytm:

1. Jeżeli na prezentowane wejście odpowiedź sieci jest prawidłowa ($a = t, e = t - a = 0$), wówczas nie zmieniamy wartości w wektora wagowego.
2. Jeżeli odpowiedzią neuronu jest 0, a powinno być 1 ($a = 0, t = 1, e = t - a = 1$), wówczas wejściowy wektor p jest dodawany do wektora wagowego. W ten sposób punkt wektora wagowego przybliża się do punktu wejściowego, zwiększając tym samym szansę na to, że w przyszłości zostanie zaklasyfikowany jako 1.
3. Jeżeli odpowiedzią neuronu jest 1, a powinno być 0 ($a = 1, t = 0, e = t - a = -1$), wówczas wejściowy wektor p jest odejmowany od wektora wagowego w , co oddala punkt wektora wagowego od wektora wejściowego, zwiększając tym samym szansę na zaklasyfikowanie go w przyszłości jako 0.

Wszystkie wymienione wyżej przypadki opisuje wzór:

$$\Delta w = (t - a)p^T = ep^T \quad (1)$$

Analogicznie można wyznaczyć wzór zmian parametru bias neuronu:

$$\Delta b = (t - a)1 = e \quad (2)$$

Dla całej warstwy neuronów:

$$\Delta W = (t - a)p^T = ep^T \quad (3)$$

$$\Delta b = (t - a) = E \quad (4)$$

Reguły nauczania perceptronu można podsumować:

$$W^{new} = W^{old} + e(p)^T \quad (5)$$

$$b^{new} = b^{old} + e \quad (6)$$

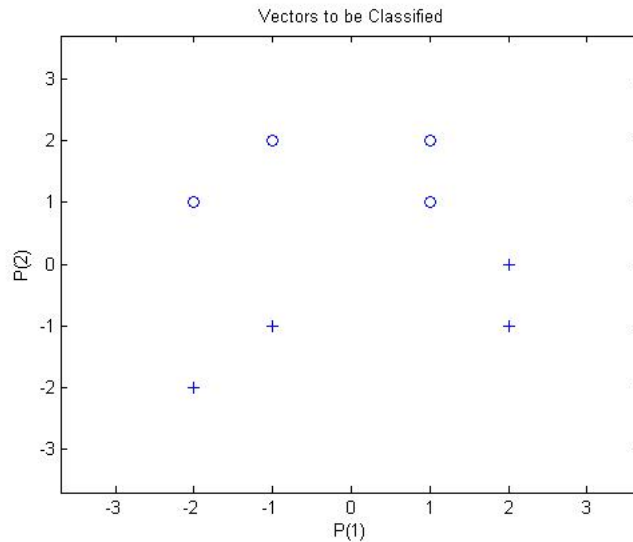
gdzie $e = t - a$.

Dokładniejsze informacje, przykładowo: <http://hagan.okstate.edu/nnd.html>

1.2 Zad. 1.

W celu rozwiązania problemu przedstawionego na rys. 1 należy zorganizować perceptron (pakiet Matlab). Wskazówki:

- Przygotowanie danych. Zapis do macierzy, przykładowo P (próbki wejściowe) oraz T (wyjścia). Należy pamiętać o wierszowym układzie, tzn. wiersze macierzy odpowiadają wejściom perceptronu. Dla macierzy P o rozm. $[R \times S]$, R to liczba wejść, S liczba próbek.
- Inicjacja perceptronu. Funkcja `newp`, gdzie przekazujemy m.in. zakresy wejść oraz wyjść (np. `net=newp(minmax(P), 1)`). Po zainicjowaniu należy sprawdzić parametry perceptronu, m.in.:
 - informacje o wagach (`net.inputweights{1}`) oraz ich wartości (`net.IW{1}`),



Rysunek 1: Klasyfikacja binarna

- informacje o biasie (`net.biases{1}`) oraz wartość (`net.b{1}`),
- zastosowana funkcja przejścia (`net.layers{1}.transferFcn`).
- Wizualizacja problemu (`plotpv(P,T)`) oraz prostej klasyfikacyjnej perceptronu (`linehandle=plotpc(net.IW{1},net.b{1});`).
- Strojenie perceptronu, instrukcje `adapt` lub `train`. Strojenie należy przeprowadzić krokowo, w celu przedstawienia jego przebiegu (pomocna funkcja `linehandle = plotpc(net.IW{1}, net.b{1}, linehandle);`).
- Weryfikacja perceptronu (instr. `sim`).

Pytania kontrolne:

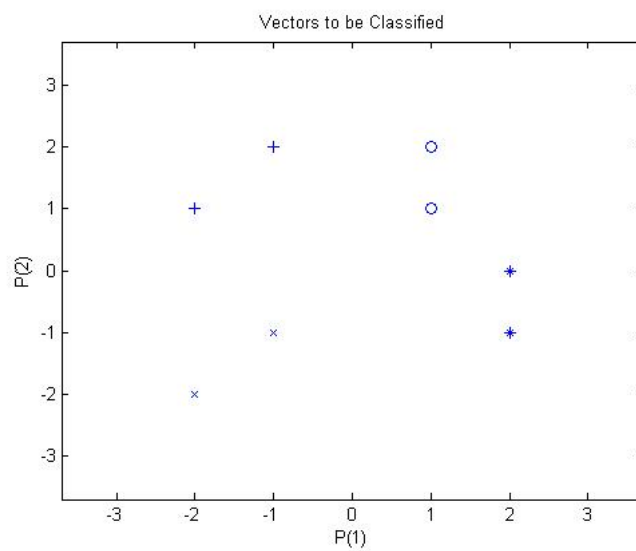
1. Oprócz standardowo zastosowanej funkcji unipolarnej, jakie inne funkcje przejścia możemy wykorzystać do perceptronu binarnego w Matlabie?
2. Za co dokładnie odpowiada funkcja `learnp` (`net.inputweights{1}.learnFcn`)?
3. Wymień inne sposoby inicjacji wag oraz biasu.

1.3 Zad. 2.

W celu rozwiązania przykładowego problemu przedstawionego na rys. 2, należy napisać uniwersalną funkcję realizującą (inicjacja, trening, weryfikacja) perceptron, który może być złożony z kilku neuronów. Argumenty wejściowe to P – macierz wejść oraz T – docelowa, liczba neuronów będzie równoważna liczbie wierszy mac. T .

1.4 Zadanie domowe

1. Jakie ograniczenia posiadają perceptrony binarne? Do rozwiązywania jakiego typu problemów mogą być stosowane?
2. Czy perceptron binarny może rozwiązać problem *XOR*? Jeżeli tak, zaproponuj odpowiednie rozwiązanie.
3. Czy potrafisz znaleźć inny rzeczywisty problem (przykładowo, z powszechnie dostępnych repozytoriów z danymi w internecie, bądź innego źródła), który można rozwiązać przy pomocy omówionej metody? Jeżeli tak, przedstaw rozwiązanie.



Rysunek 2: Problem klasyfikacji czterech klas