17.10.2017

Katarzyna Mlicka

Inżynieria Obliczeniowa

Wydział Inżynierii Metali i Informatyki Przemysłowej AGH

Nr indeksu 286120

SPRAWOZDANIE 1

**Budowa i działanie perceptronu**

Celem ćwiczenia jest poznanie budowy i działanie perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

Wstęp teoretyczny:

Perceptron – pierwsza formalna definicja sztucznego neutronu, składa się z jednego bądź wielu niezależnych neuronów McCullocha-Pittsa, implementująca algorytm uczenia nadzorowanego (uczenia z nauczycielem) klasyfikatorów binarnych. Zbudowany jest z neuronów ze skokową funkcją przejścia (unipolarną lub bipolarną):

|  |  |
| --- | --- |
| unipolarną | bipolarną |
|  |  |

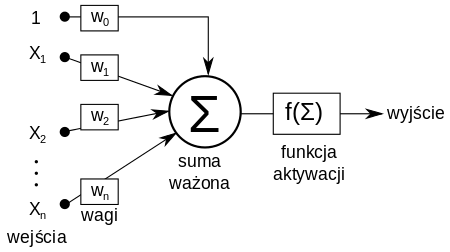
Neuron McCullocha-Pittsa to jeden z matematycznych modeli neuronu. Posiada wiele wejść i jedno wyjście. Każdemu z wejść przyporządkowana jest liczba rzeczywista - waga wejścia. Wartość na wyjściu neuronu obliczana jest za pomocą wzoru:



s = wo + xi wi

Obliczana jest suma iloczynów wartości xi podanych na wejścia i wag wi wejść.

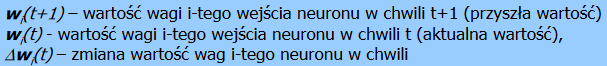
Na wyjście podawana jest wartość funkcji aktywacji f(s) dla obliczonej sumy.

Schemat neuronu McCullocha-Pittsa

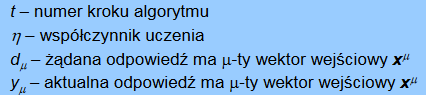
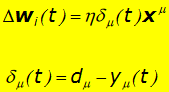
Algorytm uczenia

Uczenie perceptronu zarówno jak i innych sztucznych neuronów polega na modyfikacji wektora wag:





Najprostszą metodą uczenia jest modyfikowanie wektora wag o ustaloną wartość proporcjonalną do popełnionego błędu:



Implementacja sztucznego neuronu

>> newp

Funkcja tworzy jednowarstwową sieć neuronową, złożoną z zadanej liczby perceptronów.

>> net = newp(pr, s, tf, lf);

wejścia:

pr – macierz o rozmiarze Rx2 zawierające min i max wartości dla R wejść sieci

s – liczba neuronów w sieci

tf – funkcja aktywacyjna neuronów, domyślnie = 'hardlim'

lf – nazwa funkcji wykorzystywana do modyfikacji, domyślnie = 'learnp'

wyjścia:

net – obiekt opisujący architekturę utworzonej sieci neuronowej

>> train

Funkcja realizuje trening sieci neuronowej, wykorzystując funkcję treningu której nzawa została podana w polu net.TrainFcn zaś zartości niezbędnych parametrów w polu net.train.Param

Funkcja teningu jest uniwersalna służy do uczenia wszystkich typów sieci.

>> net = train(net, P, T);

wejścia:

net - struktura (obiekt) zawierający opis wielowarstwowej sieci jednokierunkowej

P – macierz wektorów wejściowych

T – macierz pożądanych odpowiedzi sieci

wyjście:

net – struktura (obiekt) zawierający opis nauczonej sieci jednokierunkowej

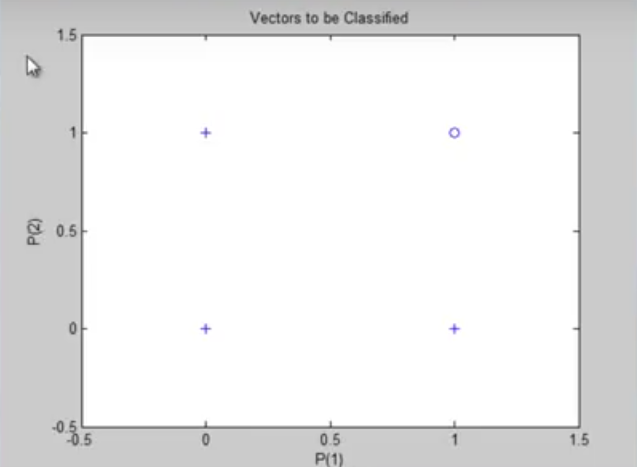
>> adapt

Funkcja realizuje jeden krok (tzn. Epokę) treningu sieci neuronowej dokonując adaptacji wag sieci. Do adaptacji jest wykorzystywana funkcja podana w polu net.adaptFcn, zaś wartość niezbędnych parametrów tej funkcji w polu net.adaptParam.

Funkcja ma taką samą postać dla wszystkich sieci. Parametry funkcji adapt są identyczne jak w przypadku funkcji train.

>> [net, y, e, Pf, Af, tr] = adapt(net, P, T, Pi, Ai)

Perceptron prosty o n-wejściach dzieli n-1 wymiarową przestrzeń na dwie półprzestrzenie. W wypadku dwuwymiarowym można równanie perceptronu można traktować jako równanie prostej. Punkty leżące ponad nią mają wartość 1, pozostałe 0. Perceptron potrafi podzielić zbiory liniowoseparowalne (nieseparowalnych nie potrafi np funkcji XOR).

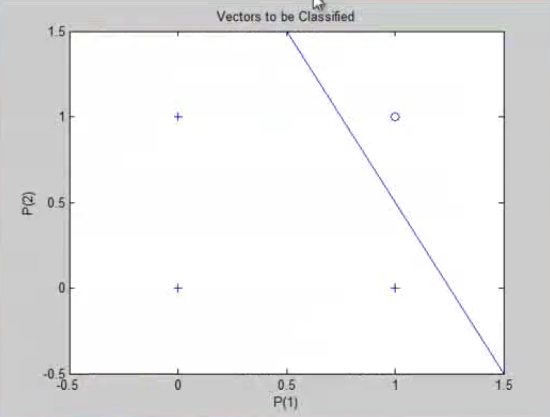
**

*>> x = [0 0 1 1; 0 1 0 1];*

*>> d = [1 1 1 0];*

*>> plotpv(x,d)*

Wykres po prawej stronie obrazuje powyższą funkcję (bramkę) logiczną NAND.

*>> siec = newp([0 1; 0 1], 1);*

*>> siec.iw{1,1} = [1 1];*

*>> siec.b(1) = 0.5;*

*>> w = siec.iw{1,1};*

*>> b = siec.b(1);*

*>> plotpv(x,d)*

*>> plotpc(w,b)*

Punkty dodatnie (czyli 'prawda') znajduje się pod kreską, natomiast ujemny ('błąd') – pod kreską. Perceptron wyłapuje błędne założenia we wprowadzonych próbach.

|  |  |
| --- | --- |
| siec = train(siec, x, d); | Testowanie błędów uczenia się. Z początku poziom błędów rośnie, następnie utrzymuje się na jednym poziomie i spada. Po spadku do zera ponowie rośnie, co może oznaczać że doszło do przetrenowania. |

Zadaniem pojedynczego perceptronu jest jedynie przetwarzanie jednostkowych informacji, podejmowanie prostych decyzji i podawanie wyników tej decyzji dalej. Dopiero wiele połączonych ze sobą węzłów jest w stanie podejmować bardziej złożone decyzje.

źródło: <http://www.neurosoft.edu.pl/media/pdf/jbartman/sztuczna_inteligencja/NTI2.pdf>