

Akademia Górniczo-Hutnicza
WIMiP, Inżynieria Obliczeniowa
grupa laboratoryjna 01
Agnieszka Kępka

26.10.2018, Kraków

Podstawy Sztucznej Inteligencji

Sprawozdanie numer 1

Budowa i działanie perceptronu

1. Cel projektu

Celem projektu było poznanie budowy i działania perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

Spis zadań do wykonania:

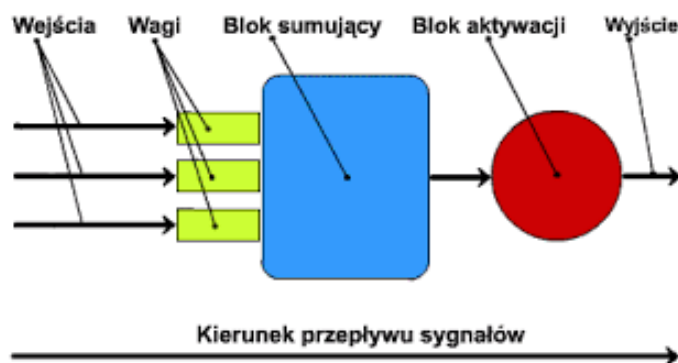
- implementacja sztucznego neuronu według podanego na wykładzie algorytmu
- wygenerowanie danych uczących i testujących wybranej funkcji logicznej dwóch zmiennych
- uczenie perceptronu dla różnej liczby danych uczących, różnych współczynników uczenia
- testowanie perceptronu

2. Zagadnienia teoretyczne

Perceptronem nazywamy prosty element obliczeniowy, który sumuje ważone sygnały wejściowe i porównuje tę sumę z progiem aktywacji - w zależności od wyniku perceptron może być albo wzbudzony (wynik 1), albo nie (wynik 0).

Sztuczny neuron - prosty system przetwarzający wartości sygnałów wprowadzanych na jego wejścia w pojedynczą wartość wyjściową, wysyłaną na jego jedynym wyjściu (dokładny sposób funkcjonowania określony jest przez przyjęty model neuronu). Jest to podstawowy element sieci neuronowych, jednej z metod sztucznej inteligencji, pierwowzorem zbudowania sztucznego neuronu był biologiczny neuron.

Schemat sztucznego neuronu:



Algorytm uczenia sieci jest procesem dochodzenia wag do wartości optymalnych – zapewniających odpowiednie reakcje sieci na pokazywane jej sygnały wejściowe, czyli prawidłowe rozwiązanie zadań.

3. Wyniki

Kod programu napisany w pakiecie Matlab:

```
net=newp([0 1; -2 2],1);  
  
W=[0 0 1 1; 0 1 0 1];  
T=[0 0 0 1];  
net=init(net);  
before=sim(net,W)  
net.trainParam.epochs=20;  
  
net=train(net,W,T);  
after=sim(net,W)
```

Powyższy kod przedstawia wywołanie funkcji, która tworzy sieć zawierającą pojedynczy neuron o dwóch wejściach.

Zakresy wartości:

- pierwszego wejścia to [0,1]
- drugiego wejścia to [-2,2]

Nastąpiło utworzenie sieci, a następnie wykonane zostało uczenie z maksymalną liczbą iteracji równą 20.

Wykorzystana została bramka AND, której koniunkcja dwóch zdań „a” i „b” jest zdaniem prawdziwym wtedy i tylko wtedy, gdy oba zdania „a”, „b” są zdaniami prawdziwymi.

(poniższy rysunek przedstawia bramkę AND)



a	b	x
0	0	0
0	1	0
1	0	0
1	1	1

Wywołanie funkcji: **NET = NEWP(PR, S, TF, LF)**, gdzie:

WEJŚCIE:

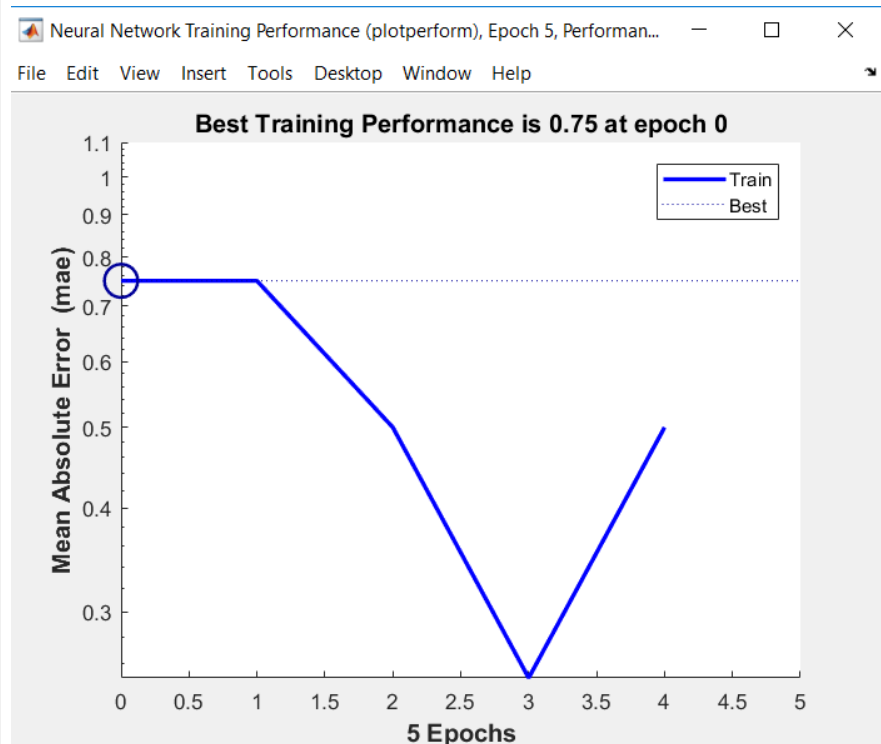
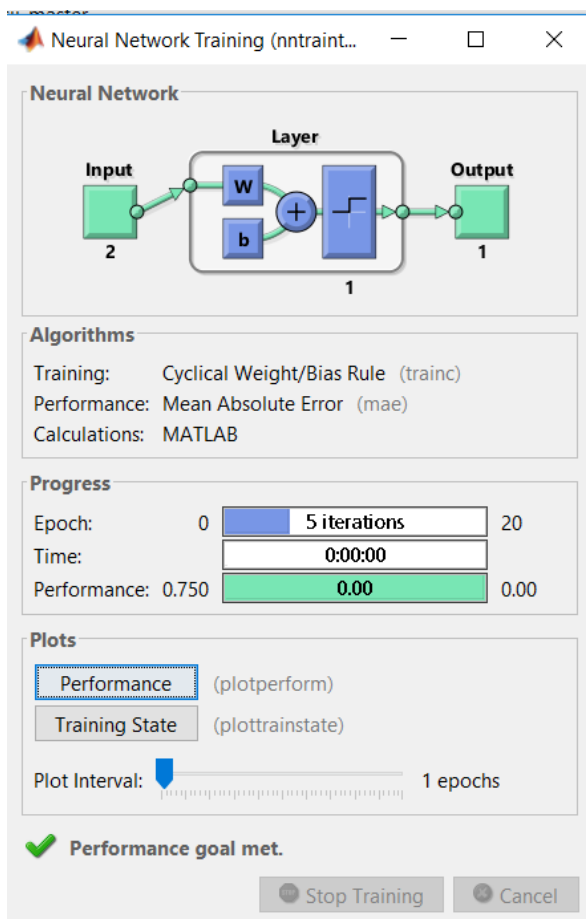
- **NEWP**- jest to funkcja, która tworzy jednowarstwową sieć neuronową, złożoną z zadanej liczby neuronów o funkcjach aktywacji „twardego” perceptronu .
- **PR**- macierz o wymiarach $R \times 2$, gdzie R jest liczbą wejść sieci, pierwsza kolumna macierzy R zawiera minimalne wartości kolejnych współrzędnych wektorów.
- **S** - liczba neuronów sieci.
- **TF** - nazwa funkcji aktywacji neuronów (zmienna tekstowa); nazwa domyślna ='hardlim'.

- **LF** - nazwa funkcji trenowania sieci perceptronowej (zmienna tekstowa); nazwa domyślna= learnp'.

WYJŚCIE:

- **NET** - struktura (obiekt) zawierająca opis architektury, metod treningu, wartości liczbowe wag i progów oraz inne parametry sieci perceptronowej.
- **W** i **T** są wartościami binarnymi, które opisują działanie bramki logicznej AND, pierwszą instrukcją jest inicjalizacja sieci perceptronowej (w tej metodzie inicjalizacji wartości wag i progów są wyznaczane w sposób losowy).
Drugą instrukcją – symulacja sieci, następnie określamy liczbę epok-w moim kodzie występuje 20 epok, wywołujemy funkcję 'train' (dokonującą treningu sieci), a następnie ponownie symulujemy działanie sieci dla wartości parametrów wyznaczonych w procesie treningu. Gdy program zostanie uruchomiony zostanie przedstawiony nam zestaw liczb 0;1 które neuron ma „przed” nauczaniem się i „po”.

Proces uczenia się sieci neuronowych przebiegł sprawnie, w piątej epoce zakończyła się nauka, wydajność wyniosła 0.750



MAE- absolutny błąd - ilość błędów w pomiarach

Wybierając opcję Plot > Performance uzyskałam wykres „uczenia się” sieci, widzimy że w epoce

zerowej MAE wynosi 0.75 jest ona stabilna do epoki pierwszej, później błąd znacznie maleje i w epoce trzeciej następuje nauczanie sieci. Widzimy jednak, że od epoki trzeciej rośnie nam MAE, oznacza to jednak, że nastąpiło „przeuczenie” sieci, jest to zjawisko polegające na nadmiernym dopasowaniu się sieci do punktów uczących, któremu towarzyszy błędne działanie sieci dla danych nie prezentowanych w trakcie uczenia. Przeuczenie pojawia się w przypadku zbyt długiego uczenia (działania algorytmu uczącego) lub wówczas, gdy zastosowana sieć jest zbyt złożona w porównaniu ze złożonością problemu lub liczbą dostępnych danych uczących.

4. Wnioski:

-Po wykonaniu projektu dowiedziałam się jak działa perceptron i algorytm uczenia. Po przetestowaniu działania programu zauważyłam, że zmiana wag wpływa na długość trwania algorytmu uczenia sieci. Wysokie wagi powodują, że czas nauki wydłuża się nawet kilkukrotnie.

-Funkcja TRAIN wygenerowała zamierzony efekt, sieć została nauczona, zamierzony cel chciałam uzyskać w 20 epokach (iteracjach) w tym przypadku sieć dokonała tego do 3 epoki, później jednak wystąpiło przeuczenie.

-Aby zwiększyć pewność poprawnego działania perceptronu potrzebuje on wprowadzenia wartości bramek logicznych, możliwe, że sieć neuronowa jest zbyt prosta, by samodzielnie przewidzieć wyniki dla takich danych.