

# Sprawozdanie z wyznaczania granicy decyzyjnej sieci perceptronowej

Wykonali: Squiddy Skeith i Szymon

Przedmiot: Metody sztucznej inteligencji i inżynieria wiedzy

Otrzymane punkty do wyznaczenia granicy decyzyjnej (14):

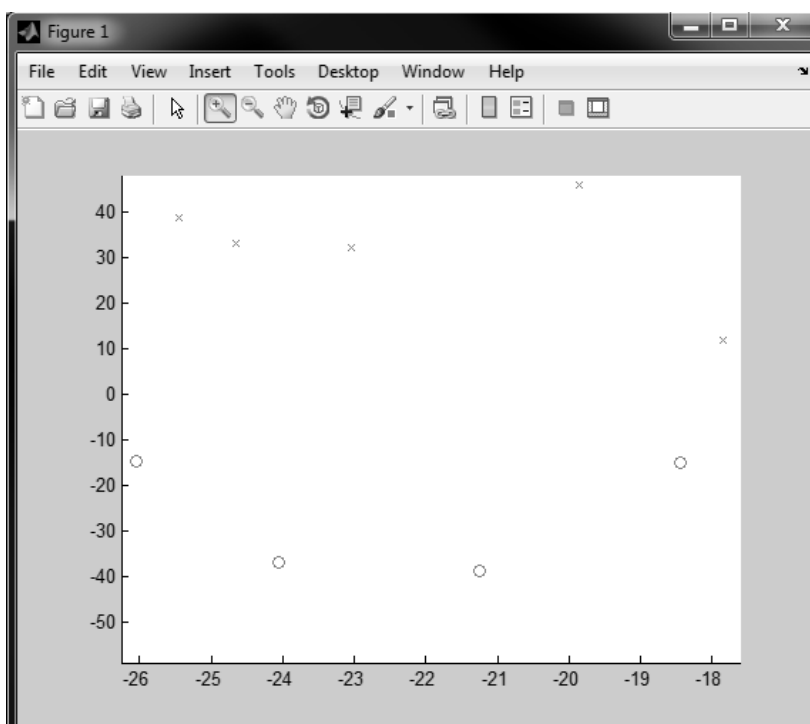
X= -25.45 -26.05 -24.65 -24.05 -23.05 -21.25 -19.85 -18.45 -17.85  
Y= 38.72 -14.52 33.04 -36.72 32.18 -38.6 45.96 -14.98 11.76

1. Przenosimy uzyskane dane do programu matlab:

```
1 %%Punkty
2 x1=[-25.45 -24.65 -23.05 -19.85 -17.85];
3 y1=[38.72 33.04 32.18 45.96 11.76];
4 x2=[-26.05 -24.05 -21.25 -18.45];
5 y2=[-14.52 -36.72 -38.6 -14.98];
6
7 %%Punkty w macierzy:
8 xy=[-25.45 -26.05 -24.65 -24.05 -23.05 -21.25 -19.85 -18.45 -17.85; 38.72 -14.52 33.04 -36.72 32.18 -38.6 45.96 -14.98 11.76];
9
10 %%Cel:
11 target=[1 0 1 0 1 0 1 0 1];
12
13 %%Wykres
14 hold on;
15 plot(x1,y1,'x')
16 plot(x2,y2,'ro')
```

2. Z podanych danych wyświetlamy wykres:

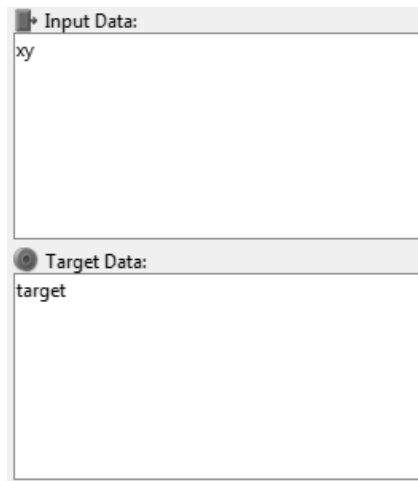
Jak widać punkty są oddzielone od siebie L1(x1,y1) oznaczone 'x' i L2(x2,y2) oznaczone 'o'.



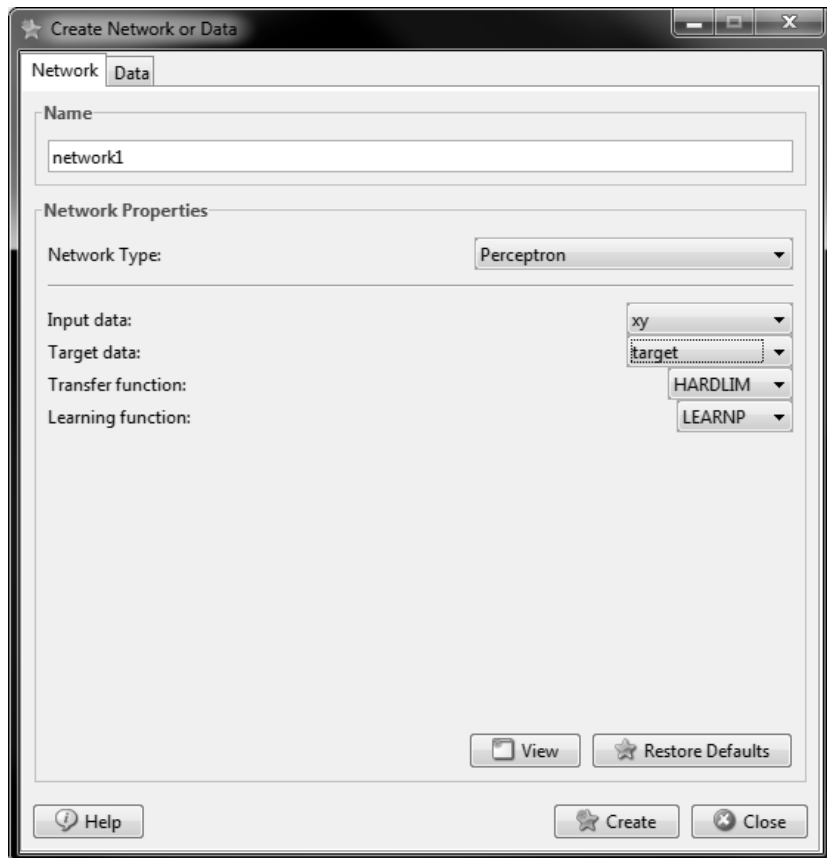
3. Włączamy nntool i importujemy nasze dane, które muszą być w workspace.

Input Data: xy

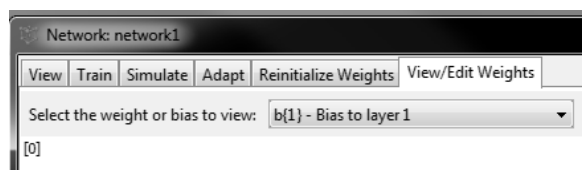
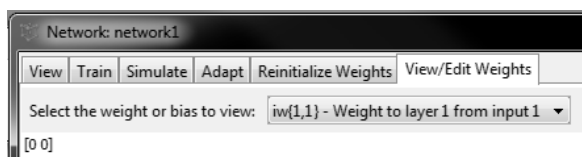
Target Data: target



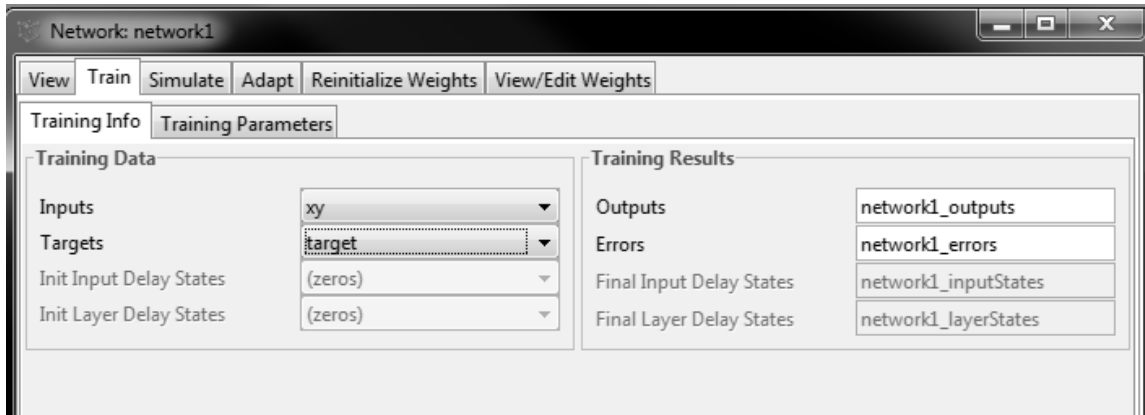
4. Po zaimportowaniu danych klikamy „New” w celu utworzenia nowej sieci perceptronowej:



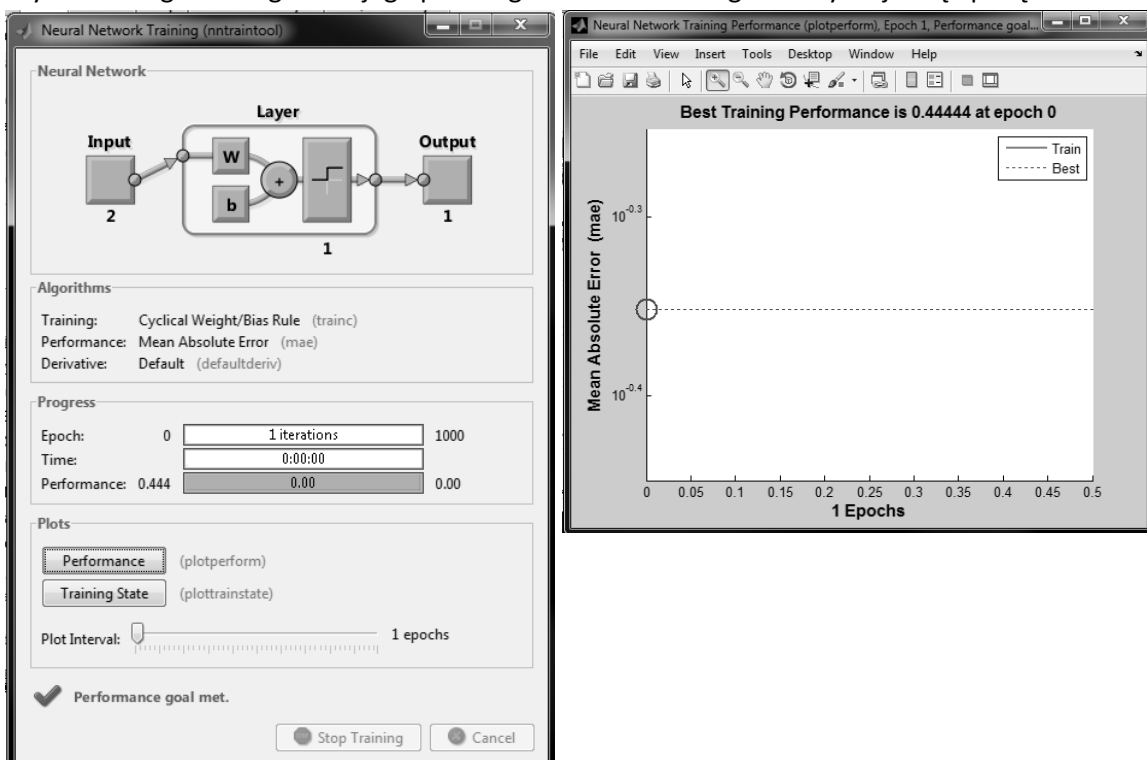
5. Nasze wagi (iw) oraz bias (b) początkowe:



6. Konfigurujemy naszą sieć w celu dokonania jej treningu:



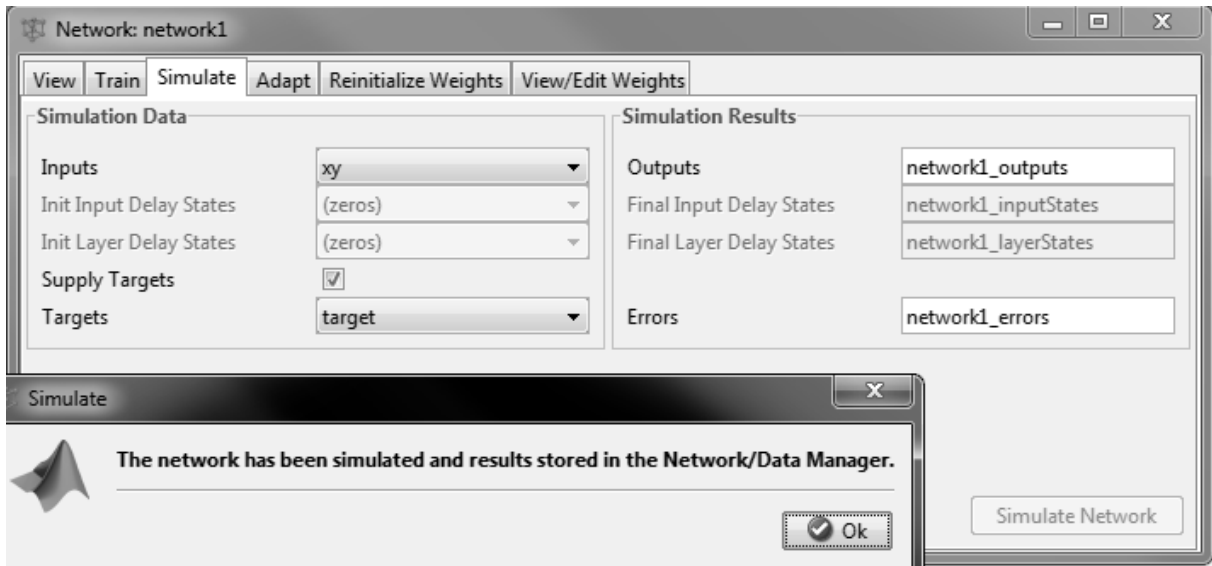
7. Wyniki naszego treningu oraz jego przebieg. Jak widać trening trwał tylko jedną epokę.



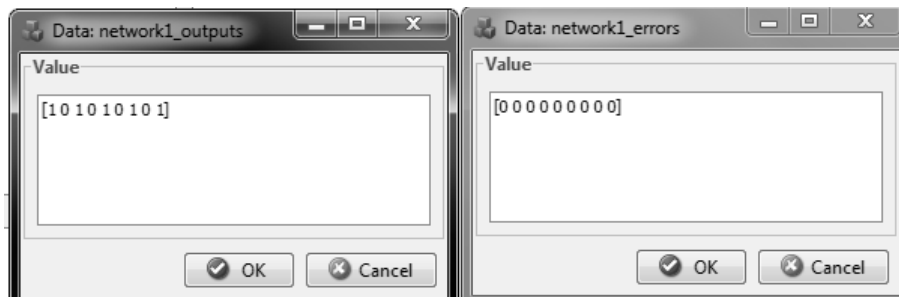
Granica decyzyjna sieci perceptronowej wynosi  $y=10^{-0.35}$  co pokazuje nam przynależność punktów według naszych założeń.

Nasze wagi (iw) zostały zmienione na [1.4 47.56], natomiast bias został bez zmian [0]

8. Symulacja sieci przeszła pomyślnie:



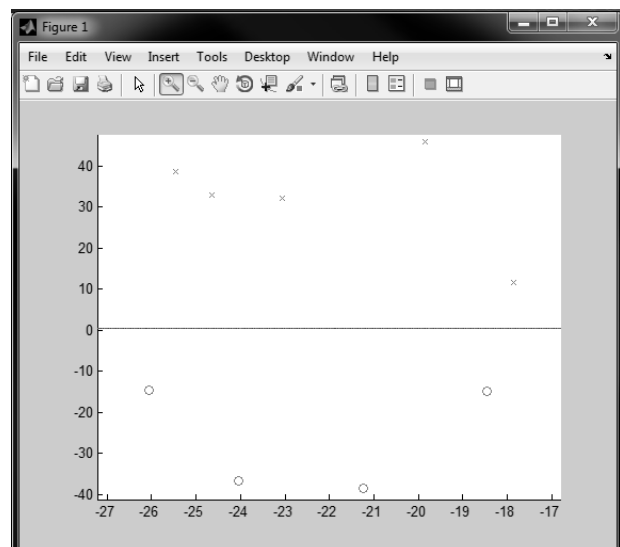
9. Dowodem na działanie naszej sieci perceptronowej jest brak błędów oraz poprawne wnioskowanie klas według naszego targetu:



10. Wykres po nałożeniu granicy wygląda następująco:

%%Granica decyzyjna:

```
x=linspace(-30,0,10000)  
y=10^-0.35;
```



11. Wnioski:

Dzięki algorytmom geometrycznych jesteśmy w stanie nauczyć komputer odróżniać jabłka według naszego wzorca. Sieć perceptronowa idealnie nadaje się do takiego klasyfikowania, ponieważ po nauczaniu naszej sieci, komputer sam zacznie odróżniać co gdzie należy zgodnie z wyznaczoną granicą decyzyjną.