

1) Opis budowy:

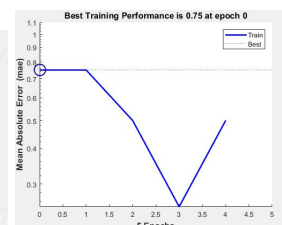
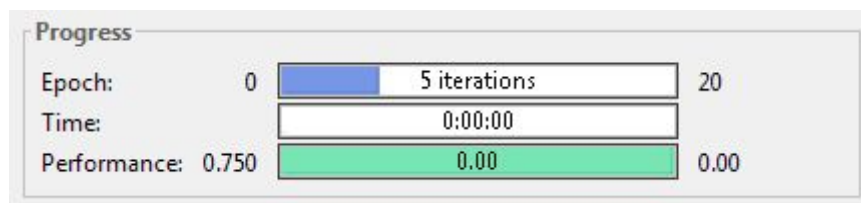
Perceptron - najprostsza sieć neuronowa, wzorowana na sposobie działania biologicznego neuronu. Zasada działania perceptronu (uczenie nadzorowane), polega na iteracyjnym dostosowywaniu wag jego elementów, tak aby przy zadanych danych wejściowych dojść do momentu w którym otrzymuje się pożądaný wynik. Perceptron zbudowany jest jedynie z warstwy wejściowej i warstwy wyjściowej, nie istnieją połączenia pomiędzy elementami tej samej warstwy. Od strony teoretycznej jest to model matematyczny który może być opisany funkcją aktywacji - unipolarną lub bipolarną, dającą wynik w zależności od wysokości sumy sygnałów wejściowych ($x = \text{SUMA}$, $y = \text{WYNIK}$).

Algorytm uczenia perceptronu - automatyczny dobór wag na podstawie napływających przykładów. W uproszczeniu:

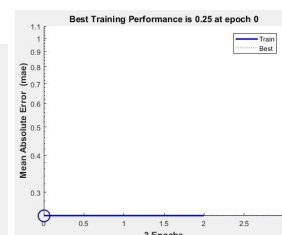
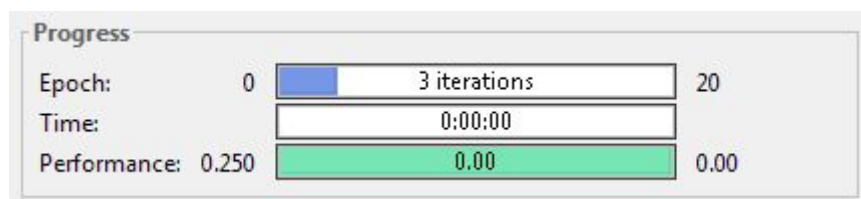
- inicjujemy wagi (np. losowo)
- dla każdego przykładu obliczamy odpowiedź
- jeśli jest nieprawidłowa to modyfikujemy wagi

2) Zestawienie otrzymanych wyników

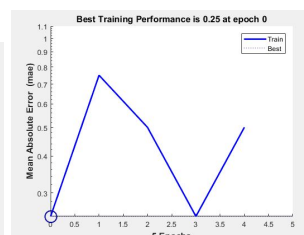
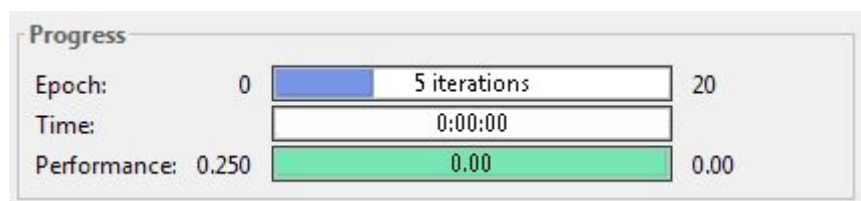
bramka AND



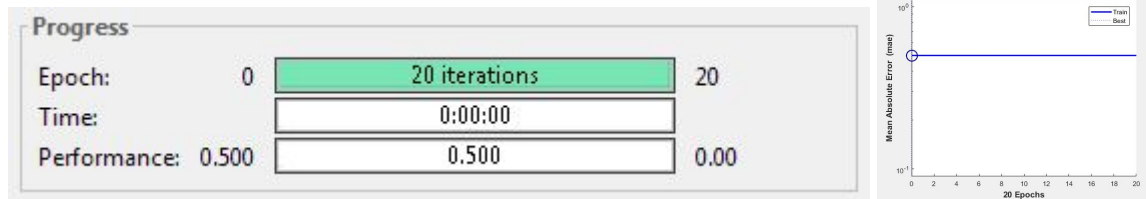
bramka OR



bramka NAND



*bramka XOR



3) Analiza wyników:

Sieć bardzo szybko radzi sobie z zadanymi problemami, bramki AND i NAND nauczyła się już po 5. iteracjach, a bramki OR już po 3.. Bramka XOR natomiast jest niemożliwa do nauczenia się przez tak prostą sieć - wynika to z faktu, że dla tego problemu, sygnały wejściowe nie są liniowo separowalne. Widzimy w tym momencie sens ograniczenia maksymalnej ilości epok.

4) Wnioski:

Model prezentuje sposób działania sieci neuronowych oraz tego w jaki sposób komputer jest w stanie się uczyć.

5) Listing kodu:

```
close all; clear all; clc;
```

```
%dwuelementowy perceptron  
net=newp([0 1; -2 2],1);
```

```
W=[0 0 1 1; 0 1 0 1];  
T=[0 0 0 1];
```

```
%inicjalizacja domyślna  
net=init(net);
```

```
sym_przed=sim(net,W)
```

```
%max ilość epok - iteracji  
net.trainParam.epochs=20;  
net=train(net,W,T);
```

```
sym_po = sim(net,W)
```