

# Podstawy Sztucznej Inteligencji

## Sprawozdanie - Scenariusz nr.1

Tomasz Lenart, gr. 2

### Cel ćwiczenia:

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

### Przebieg ćwiczenia:

- Zaimplementowano sztuczny perceptron przy pomocy framework'u Encog.
- Dobrano uczące i testujące dane. W celu sprawdzenia działania programu wybrano operację logiczną XOR, której tablica prawdy wygląda następująco:

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

1 XOR 1 = 0

Dzięki niej dobrano odpowiednie dane wejściowe, które przechowywano w tablicach.

```
public static double[][] XOR_INPUT = {  
    new double[2] { 0.0, 0.0 },  
    new double[2] { 1.0, 0.0 },  
    new double[2] { 0.0, 1.0 },  
    new double[2] { 1.0, 1.0 } };
```

Z powyższej tabeli wyznaczono również możliwe wyniki operacji XOR:

```
public static double[][] XOR_IDEAL = {  
    new double[1] { 0.0 },  
    new double[1] { 1.0 },  
    new double[1] { 1.0 },  
    new double[1] { 0.0 } };
```

Powyższe dwie tabele zostały wykorzystane do stworzenia zestawu uczących danych

```
INeuralDataSet trainingSet = new BasicNeuralDataSet(XOR_INPUT, XOR_IDEAL);
```

Po przygotowaniu odpowiednich danych utworzono sieć neuronową składającą się z trzech warstw:

```
BasicNetwork network = new BasicNetwork();

network.AddLayer(new BasicLayer(new ActivationSigmoid(), true, 2));
network.AddLayer(new BasicLayer(new ActivationSigmoid(), true, 6));
network.AddLayer(new BasicLayer(new ActivationSigmoid(), true, 1));
network.Structure.FinalizeStructure();
network.Reset();
```

Zostały utworzone trzy warstwy za pomocą metody AddLayer(). Pierwsza z nich to warstwa wejściowa, zawierająca dwa neurony. Ostatnia z nich to warstwa wyjściowa która zawiera pojedynczy neuron. Warstwa po środku to warstwa ukryta.

Po utworzeniu sieci należy utworzyć obiekt uczący. Encog dostarcza wielu wbudowanych funkcji odpowiadających za uczenie neuronu. Niektóre są zależne od typów neuronów, a inne np. od typów danych. W tym przypadku został użyty obiekt 'najbardziej ogólnego przeznaczenia' nazywany 'resilient propagation trainer'.

```
ITrain train = new ResilientPropagation(network, trainingSet);
```

Po utworzeniu obiektu uczącego przechodzimy przez pętlę iteracji, która ma na celu nauczenie neuronu funkcji XOR. W każdej iteracji sprawdzany jest błąd.

```
do
{
train.Iteration();
    Console.WriteLine("Iteracja: #" + iteracja + " Błąd:" + train.Error);
    iteracja++;
} while ((iteracja < 5000) && (train.Error > 0.001));
```

Program działa dopóki błąd nie jest mniejszy od 1%. Błąd określa jak wynik neuronu jest daleki od wyniku idealnego definiowanego przed rozpoczęciem uczenia.

Uruchomienie programu dało wyniki zamieszczone poniżej:

```
Iteracja#39 Error:0,050307474527291
Iteracja#40 Error:0,0334430418581486
Iteracja#41 Error:0,0205153853168482
Iteracja#42 Error:0,0112653621759788
Iteracja#43 Error:0,00548907879852843
Iteracja#44 Error:0,00232484617531865
Iteracja#45 Error:0,00083648524132924
Wyniki:
0,0, aktualne=2,32808498421661E-05,idealne=0
1,0, aktualne=0,999848557883492,idealne=1
0,1, aktualne=0,999920417520307,idealne=1
1,1, aktualne=0,000468510901040711,idealne=0
```

```
Iteracja#30 Error:0,0497334198668544
Iteracja#31 Error:0,0321013223649927
Iteracja#32 Error:0,0245825629567189
Iteracja#33 Error:0,021288127740029
Iteracja#34 Error:0,0152989908380014
Iteracja#35 Error:0,0115885774904528
Iteracja#36 Error:0,00934296101033169
Iteracja#37 Error:0,00578824778548477
Iteracja#38 Error:0,00329470555021435
Iteracja#39 Error:0,00169395778385008
Iteracja#40 Error:0,000780076285129729
Wyniki:
0,0, aktualne=6,22419548092264E-06,idealne=0
1,0, aktualne=0,999351801714572,idealne=1
0,1, aktualne=0,999995678253614,idealne=1
1,1, aktualne=4,94526839709452E-05,idealne=0
```

```
Iteracja#40 Error:0,177284095831158
Iteracja#41 Error:0,145698986214678
Iteracja#42 Error:0,114429674447496
Iteracja#43 Error:0,0847096655870408
Iteracja#44 Error:0,0582020882987324
Iteracja#45 Error:0,0366208714103808
Iteracja#46 Error:0,0208886246445533
Iteracja#47 Error:0,0107150015134239
Iteracja#48 Error:0,00486078532539344
Iteracja#49 Error:0,00190780169265693
Iteracja#50 Error:0,000632196080973561
Wyniki:
0,0, aktualne=1,03312799937878E-05,idealne=0
1,0, aktualne=0,999762746584404,idealne=1
0,1, aktualne=0,999941449081298,idealne=1
1,1, aktualne=0,000230010017935711,idealne=0
```

**Podsumowanie i wnioski:**

Po zakończeniu działania programu widzimy aktualne wartości na wyjściach neuronu. Wartości na wyjściu nie są jednak równe co do wartości idealnych ale było to do przewidzenia. Mimo tego wyniki są bardzo bliskie do wartości oczekiwanych. Wyniki z trzech uruchomień programu różnią się od siebie, jednak ma to związek z różnymi wagami.

Na podstawie tych wyników możemy wnioskować że neuron jest w stanie pracować na odpowiednich wartościach tak jak funkcja XOR.