Barbara Partyka
Scenariusz 2
Temat ćwiczenia: Budowa i działanie sieci jednowarstwowej
Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

Sieć neuronowa– ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu. (zródło:https://pl.wikipedia.org/wiki/Sie%C4%87_neuronowa)

Uproszczony schemat jednokierunkowej sieci neuronowej. Poszczególne "kółka" oznaczają sztuczne neurony.

Problem zrealizowano za pomocą jednego perceptronu. Przygotowano dane uczące zawierające 10 małych liter oraz 10 dużych liter w postaci siatki 5x7 w reprezentacji zero-jedynkowej.
Wykorzystano algorytm delta.

DELTA, zakłada że w raz z każdym wektorem wejściowym X do neuronu podawany jest sygnał z. Neuron odpowiada na sygnał X sygnałem

$$y = W * X$$

Przy czym jeśli neuron nie jest nauczony, sygnał ten jest inny niż wymagany ($y \neq z$). Wewnątrz neuronu istnieje blok oceniający wielkość błędu

$$\delta = z - y$$

Blok ten składa się inwertora oraz sumatora. Na podstawie sygnału błedu oraz wektora wejściowego X możliwe jest takie skorygowanie wektora wag W, by neuron lepiej realizował zadaną funkcję y= f(X). Nowy wektor wag W'obliczany jest ze wzoru:

$$W' = W + \eta\delta X$$

Gdzie η jest współczynnikiem liczbowym, decydującym o szybkości nauki.

(źródło: http://galaxy.agh.edu.pl/~vlsi/AI/rdelta/)

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

Możemy do tego użyć algorytmu uczenia perceptronu- tzn. automatycznego doboru wag na podstawie napływajacych przykładów. W uproszczonym (ze względu na ilustrację) przypadku dwuwymiarowym algorytm wygląda następująco:

- Inicjujemy wagi losowo.
- Dla każdego przykładu uczącego obliczamy odpowiedź perceptronu.
- Jeśli odpowiedź perceptronu jest nieprawidłowa, to modyfikujemy wagi:

w_1+=n *(d-y)* x_1

w_2+=n* (d-y)* x_2

b +=n *(d-y)

Program testowano dla 3 współczynników uczenia: 0,1; 0,01; 0,001.

Zawartość pliku trainData.txt stosowany do nauki perceptronu:

```
1
0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 1 0 1 1 1 1
1
1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0
0
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0
1
0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0
1
1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0
0
```

```
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1 1
1
1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0
1
1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0
0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0
1
0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0
0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0
1
1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
0
1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
1
0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0
0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0
1
1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0
0
0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0
```

Zawartość pliku testData.txt testującego neuron:

```
1
```

```
0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 0 0 1 0 1 1 1 1
1
1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0
1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
0
0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0
1
0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0
1
1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0
0
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0
1
0 1 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0
1
0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0
0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0
1
1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0
0
0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0
1
1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0
0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1 1
```

```
1
1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1
0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0
0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0
1
1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1
```

Siatka małych liter -stworzona:

a

```
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
0 0 0 0 1
0 1 1 1 1
1 0 0 0 1
0 1 1 1 1
b
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 1 1 0
1 1 0 0 1
1 0 0 0 1
1 1 1 1 0
c
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 1
0 1 1 1 0
d
0 0 0 0 1
0 0 0 0 1
0 0 0 0 1
0 1 1 0 1
1 0 0 1 1
1 0 0 0 1
0 1 1 1 1
e
```

0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 0 0 0 1
1 1 1 1 1
1 0 0 0 0
0 1 1 1 0
f
0 0 1 1 0
0 1 0 0 1
0 1 0 0 0
1 1 1 0 0
0 1 0 0 0
0 1 0 0 0
0 1 0 0 0
g
0 0 0 0 0
0 1 1 1 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 1
0 0 0 0 1
0 1 1 1 0
h
1 0 0 0 0
1 0 0 0 0
1 0 1 1 0
1 1 0 0 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
i
0 0 0 0 0
0 0 1 0 0
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
j
0 0 0 1 0
0 0 0 0 0
0 0 1 1 0
0 0 0 1 0
0 0 0 1 0

```
1 0 0 1 0
0 1 1 0 0
k
1 0 0 0 0
1 0 0 0 0
1 0 0 1 0
1 0 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
l
0 1 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 1 0
```

Wnioski, analiza błędów:

Na działanie sieci maja wpływ współczynnik uczenia, ilość danych uczących, funkcja aktywacji oraz metoda uczenia.

Proces nauki przebiegał szybko.

Program jest w stanie uzyskać 100% skuteczność.

Na działanie sieci maja wpływ współczynnik uczenia, ilość danych uczących, funkcja aktywacji oraz metoda uczenia.

Listing kodu:

https://github.com/barbarapar/PSI_GCP03_zima_2017-2018_Barbara_Partyka