Małgorzata Bień IS gr. 1

Sprawozdanie do scenariusza nr 6

Temat ćwiczenia: Budowa i działanie sieci Kohonena dla WTM.

1. Cel: Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTM do odwzorowywania istotnych cech liter alfabetu.

2. Syntetyczny opis budowy sieci oraz algorytmu uczenia

- Sieci Kohonena są jednym z podstawowych typów sieci samoorganizujących się.
- Podstawę samoorganizacji sieci neuronowych stanowi prawidłowość, że globalne uporządkowanie sieci jest możliwe przez działania samoorganizujące prowadzone lokalnie w różnych punktach sieci, niezależnie od siebie.
- W wyniku przyłożonych sygnałów wejściowych następuje w różnym stopniu aktywacja neuronów, dostosowująca się wskutek zmiany wartości wag synaptycznych do zmian wzorców uczących.
- W procesie uczenia istnieje tendencja do wzrostu wartości wag, dzięki której tworzy się rodzaj dodatniego sprzężenia zwrotnego: większe sygnały pobudzające, większe wartości wag, większa aktywność neuronów. Następuje przy tym naturalne zróżnicowanie wśród grup neuronów
- Uczenie odbywa się metodą samoorganizującą typu konkurencyjnego. Polega ona na podawaniu na wejścia sieci sygnałów, a następnie wybraniu w drodze konkurencji zwycięskiego neuronu, który najlepiej odpowiada wektorowi wejściowemu.

Zasady działania sieci Kohonena:

- w sieciach Kohonena, przeważnie jednowarstwowych, każdy neuron połączony jest ze wszystkimi składowymi wektora wejściowego x,
- wagi połączeń synaptycznych neuronów tworzą wektor w,
- każdy węzeł oblicza swój poziom aktywacji jako iloczyn wektora wag i wektora wejściowego
- przy założeniu normalizacji wektorów wejściowych po pobudzeniu sieci wektorem x zwycięża we współzawodnictwie neuron, którego wagi najmniej różnią się od odpowiednich składowych tego wektora (neuron, który dla danego wektora wejściowego ma najwyższy poziom aktywacji)

  Zwycięzca, neuron w-ty, spełnia relacje: $$d(\mathbf{x}, \mathbf{w}_w) = \min_{1 \leqslant i \leqslant n} d(\mathbf{x}, \mathbf{w}_i)$$

  gdzie d(x,w) oznacza odległość w sensie
  wybranej metryki między wektorem x i wektorem w, a n jest liczbą neuronów.

- W uczeniu stosuje się strategię WTA(Winner Takes All) lub użytą w projekcie WTM(Winner Takes Most)
- w strategi WTM wokół neuronu zwycięzcy przyjmuje się sąsiedztwo S(k) o określonym promieniu malejącym w czasie
- neuron zwycięzca i wszystkie neurony położone w obszarze sąsiedztwa podlegają adaptacji, zmieniając swoje wektory wag w kierunku wektora x, zgodnie z regułą Kohonena:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta_i(k)[\mathbf{x} - \mathbf{w}_i(k)]$$

gdzie η(k) jest współczynnikiem uczenia neuronu z sąsiedztwa S(k) w k-tej chwili.

- W algorytmie Kohonena funkcja sąsiedztwa G(i,x) definiowana jest w postaci:

$$G(i,\mathbf{x}) = \begin{cases} 1 & \text{dla } d(i,w) \leqslant \lambda \\ 0 & \text{dla pozostałych} \end{cases}$$

lub

$$G(i,\mathbf{x}) = \exp\left(-\frac{d^2(i,w)}{2\lambda^2}\right)$$

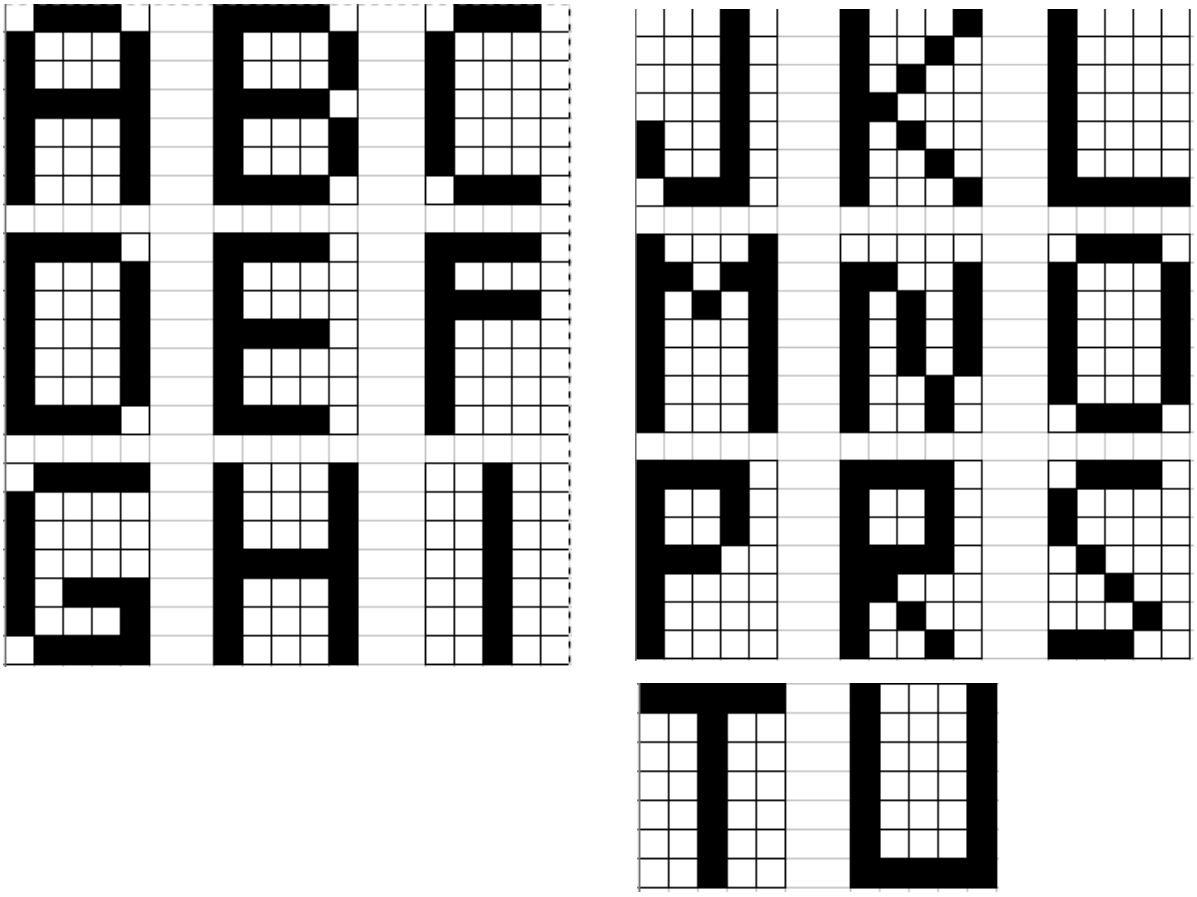
gdzie d(i,w) oznacza odległość euklidesową między wektorami wag neuronu zwycięzcy w i neuronu i-tego lub odległość mierzną w liczbie neuronów.
Współczynnik λ jest promieniem sąsiedztwa o wartości malejącej z czasem uczenia do zera.

Miara euklidesowa odległości $$d(\mathbf{x},\mathbf{w}_i) = \|\mathbf{x}-\mathbf{w}_i\| = \sqrt{\sum_{j=1}^{N}(x_j - w_{ij})^2}$$

Pierwsze sąsiedztwo nazywane jest prostokątnym, drugie gaussowskim. Sąsiedztwo gaussowskie prowadzi zwykle do lepszych rezultatów uczenia i lepszej organizacji sieci.

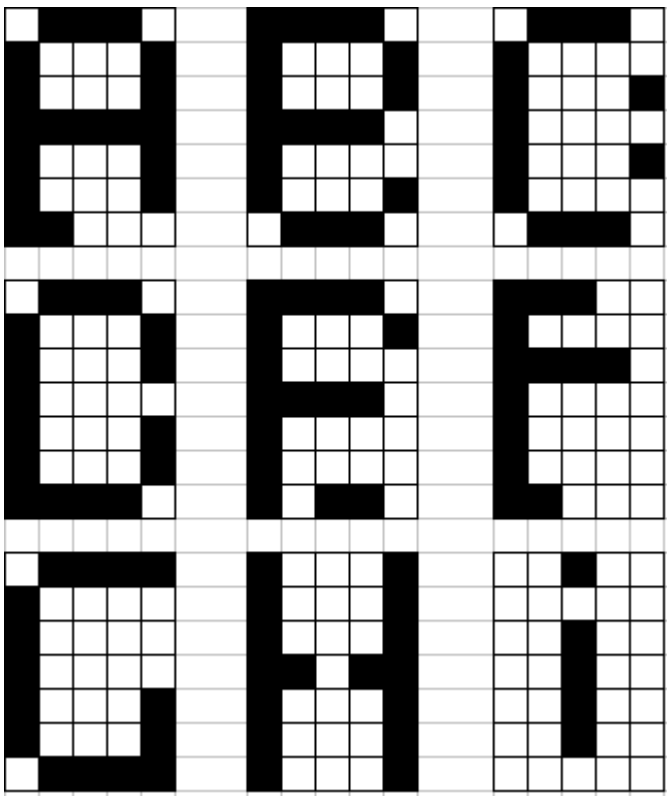
3. Dane uczące:

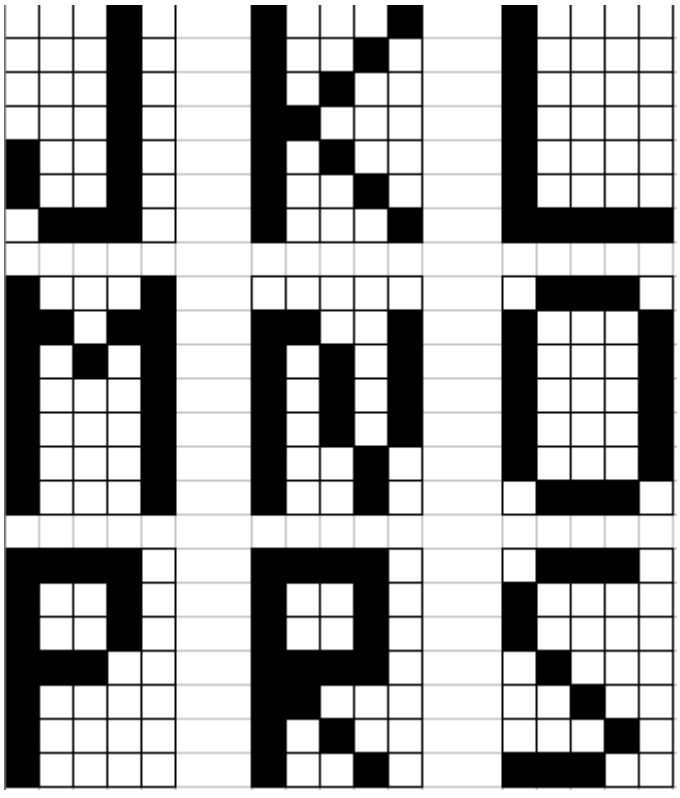
Dane uczące to duże litery alfabetu od A-U przedstawione w formacie dwuwymiarowej tablicy 5x7. Do przetworzenia w sieci zostały zapisane jako ciąg 35 cyfr 0 lub 1, które są danymi wejściowymi, plus 20 cyfr – prawidłowe dane wyjściowe (1 określająca daną literę i 19 zer).

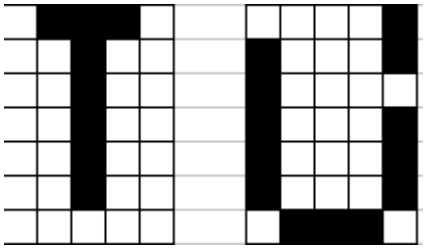


```
01110100011000111111100011000110001
11110100011000111110100011000111110
01110100001000010000100001000001110
11110100011000110001100011000111110
11110100001000011110100001000011110
11110100001111010000100001000010000
01111100001000010000101111000101111
10001100011000111111100011000110001
00100001000010000100001000010000100
00010000100001000010100101001001110
10001100101010011000101001001010001
10000100001000010000100001000011111
10001110101010110001100011000110001
00000110011010110101101011001010010
01110100011000110001100011000101110
11110100101001011100100001000010000
11110100101001011110110001010010010
01110100001000001000001000001011100
11111001000010000100001000010000100
10001100011000110001100011000111111
```

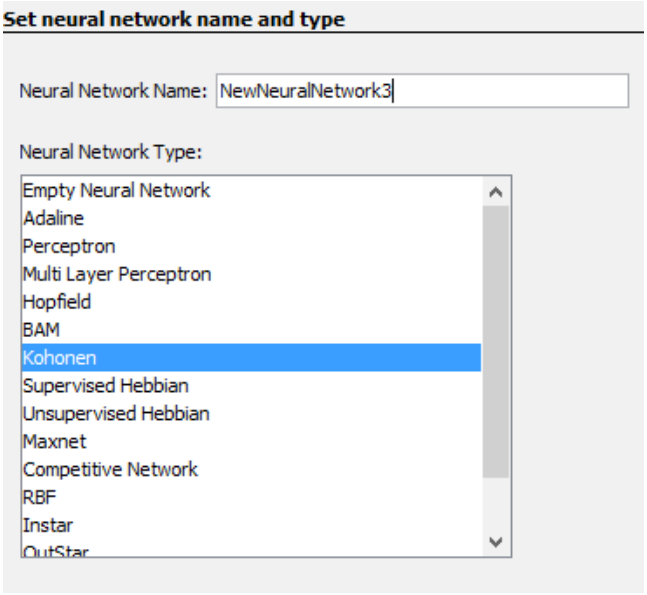
4. Dane testujące
W zestawie danych testujących zmieniłam część pikseli by sprawdzić w jakim stopniu sieć rozpoznaje daną literę.

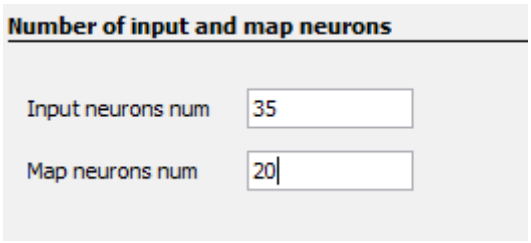


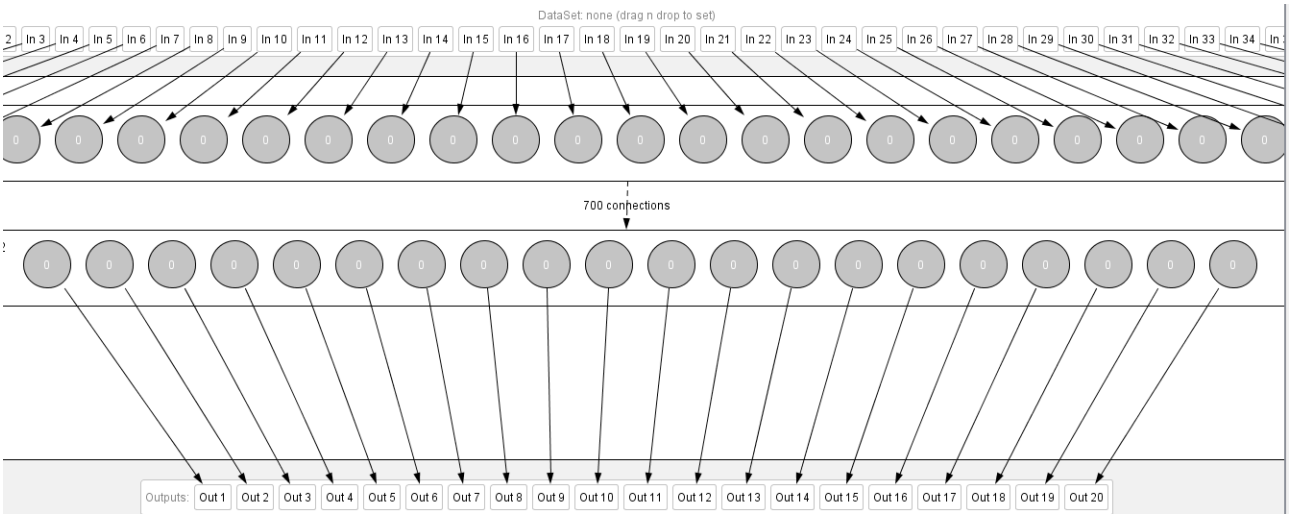




5. Konfiguracja programu



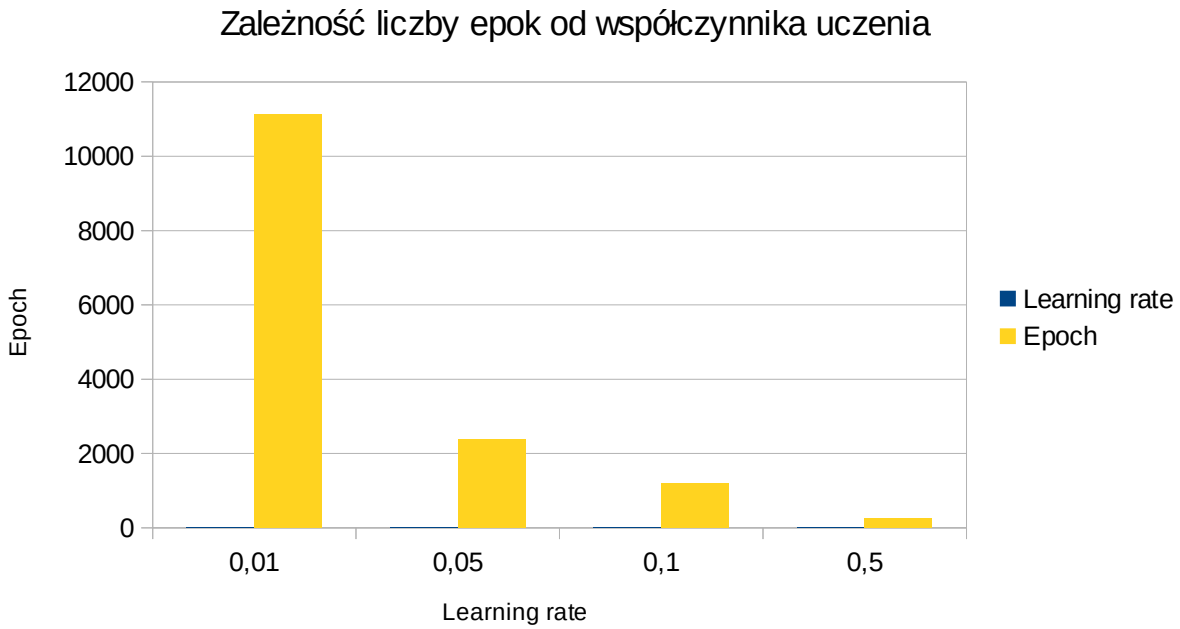


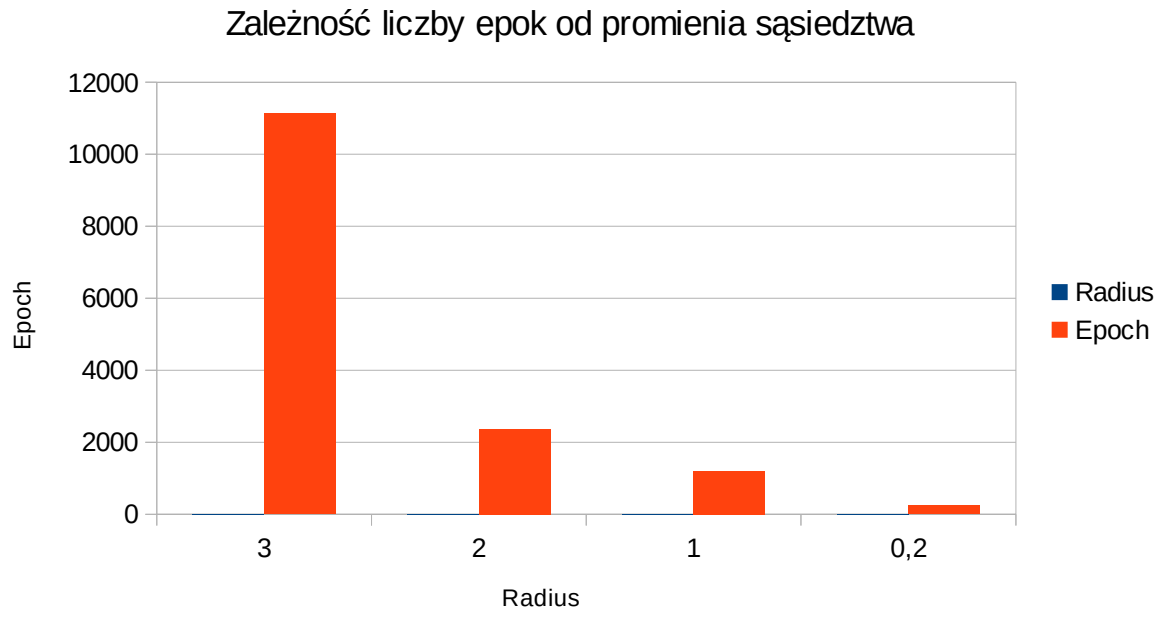


Sieć:
35 neuronów wejściowych
20 wyjść sieci

6. Wyniki
Wyniki zamieściłam w arkuszu kalkulacyjnym.

7. Analiza





Learning rate	Radius	Epoch	Error
0,01	3	11136	9.991257869684698E-4
0,05	2	2365	9.956339521570634E-4
0,1	1	1205	9.925464140856197E-4
0,5	0,2	243	9.637539346035538E-4

Sieć uczyłam z wykorzystaniem czterech współczynników uczenia i promieni sąsiedztwa. Najwolniej proces uczenia następował przy najmniejszym Learning rate tj. 0,01 i było to 11136 epok. Wraz ze wzrostem współczynnika uczenia oraz zmniejszaniem promienia sąsiedztwa liczba epok potrzebna do uczenia sieci malała. We wszystkich przypadkach otrzymałam podobny błąd średniokwadratowy, minimalnie mniejszą wartość błędu uzyskałam przy uczeniu z największym współczynnikiem uczenia tj. 0,5 i najmniejszym promieniem sąsiedztwa.

Podczas uczenia zawsze otrzymywałam poprawne wyniki, tj. poprawne wskazanie litery. Do testowania użyłam zestawu testującego, w którym część pikseli była inna, zmodyfikowany był wygląd litery. W żadnym z testów sieć nie wykazała 100% poprawności działania. Błędne wskazania sieci podczas testów stanowiły od 10%-25%. Niepoprawne rozpoznanie dotyczyło liter:

- TEST 1: A,C,G,T
- TEST 2: D,E,G
- TEST 3: D,G
- TEST 4: D,G
- TEST 5: C,D,G,T

Litera A była rozpoznana jako litera B w jednym teście. Litera C była rozpoznawana jako litera O w 2 testach. Litera D była rozpoznawana jako litera O w każdym z testów. Litera G była błędnie rozpoznana w każdym z testów, w 3 przypadkach jako litera B i w 2 jako litera O. Litera T była rozpoznawana jako litera I w dwóch testach. Litera E została rozpoznana jako litera R w jednym teście. Błąd średniokwadratowy podczas testów wyniósł średnio 0,017842.

8. Wnioski

Uczenie sieci neuronowej przebiegało poprawnie. Sieć prawidłowo rozpoznaje emotikony, wynik ich rozpoznania dąży do 1, niezależnie od współczynnika uczenia i promienia sąsiedztwa. Od współczynnika zależy prędkość uczenia. Z wykresu zależności liczby epok można wnioskować, że uczenie przebiegało szybciej wraz ze wzrostem współczynnika uczenia oraz ze spadkiem wartości promienia sąsiedztwa. Przeprowadzenie testów na zestawie testującym, który zawierał zmodyfikowane litery pokazało błędy sieci w rozpoznawaniu liter. Nie wskazuje to jednak na jej nieprawidłowe działanie lecz na umiejętność rozpoznania różnych cech nauczonych liter. Sieć wskazywała błędne rozpoznanie dla liter, których wygląd był zbliżony do innych. Przykładowo litera D z zestawu testującego przypomina literę O z zestawu uczącego, dlatego nie można stwierdzić, że sieć podała niepoprawny wynik.