Sieci Kohonena (KOH)

Bartłomiej Wójcik

Numer indeksu: 327327

Spis treści

Implementacja sieci Kohonena

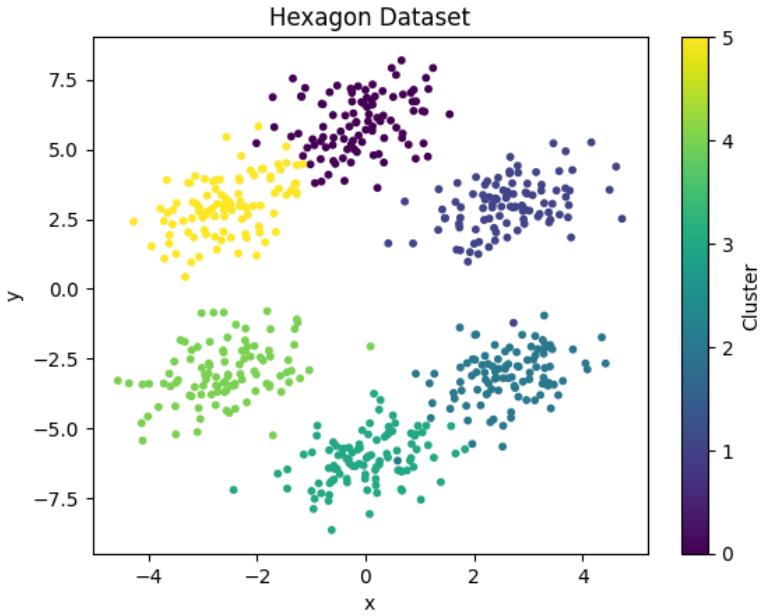
KOH1: Podstawowa sieć Kohonena

— **Cel**: Implementacja podstawowej wersji sieci Kohonena z możliwością wyboru funkcji sąsiedztwa i topologii prostokątnej.
— **Kluczowe funkcjonalności**:

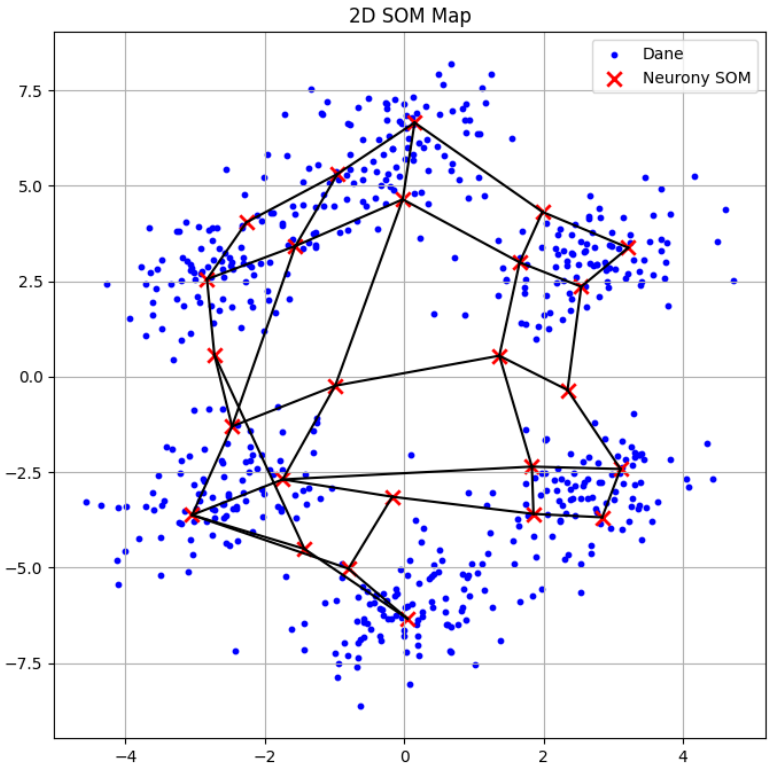
```
class SelfOrganizingMap:
    def __init__(self, grid_shape, input_dim,
                 topology='rectangular',
                 neighborhood_fn='gaussian'):
        # Inicjalizacja wag neuronów
        self.weights = np.random.randn(*grid_shape, input_dim)

    def train(self, X, num_epochs, beta, lambda_):
        # Algorytm uczenia:
        # 1. Dla każdej epoki oblicz współczynnik uczenia
        # 2. Dla każdego wzorca znajdz BMU
        # 3. Zaktualizuj wagi używając funkcji sąsiedztwa
        #    (gaussowskiej lub "mexican hat")
```

— **Testowane funkcje sąsiedztwa**:
— Funkcja gaussowska: $h(t) = e^{-\frac{||r_i - r_j||^2}{2\sigma(t)^2}}$
— "Mexican hat": $h(t) = (1 - \frac{||r_i - r_j||^2}{\sigma(t)^2})e^{-\frac{||r_i - r_j||^2}{2\sigma(t)^2}}$

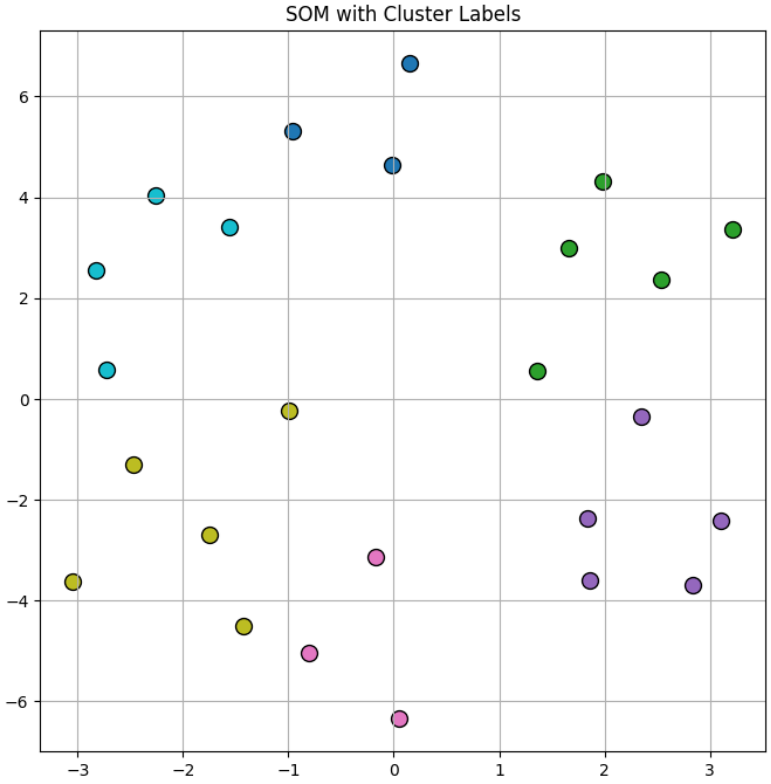
Wyniki dla hexagon dataset



Rys. 1. Wizualizacja zbioru hexagon.



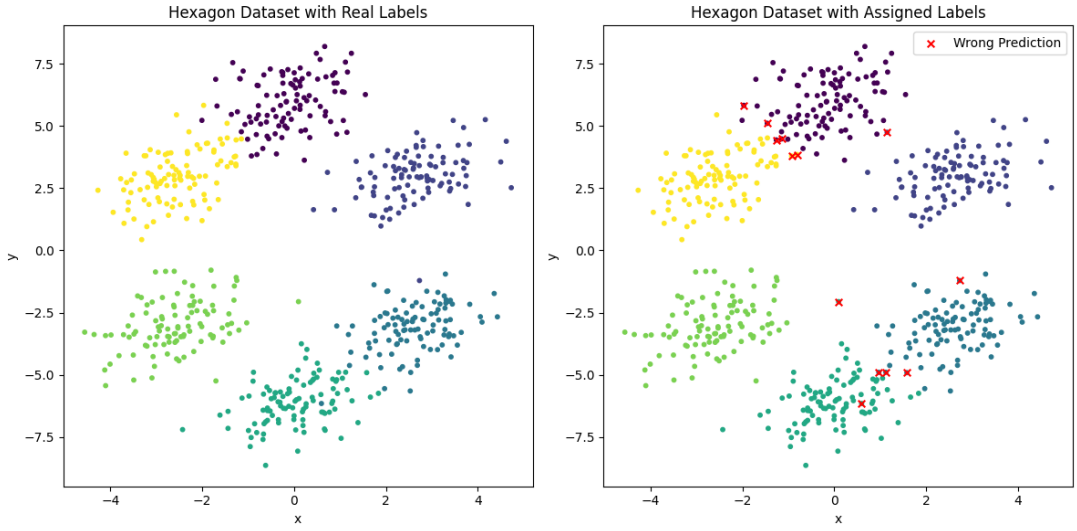
Rys. 2. Wizualizacja sieci Kohonena po trenowaniu (sieć 5x5, 100 epok).



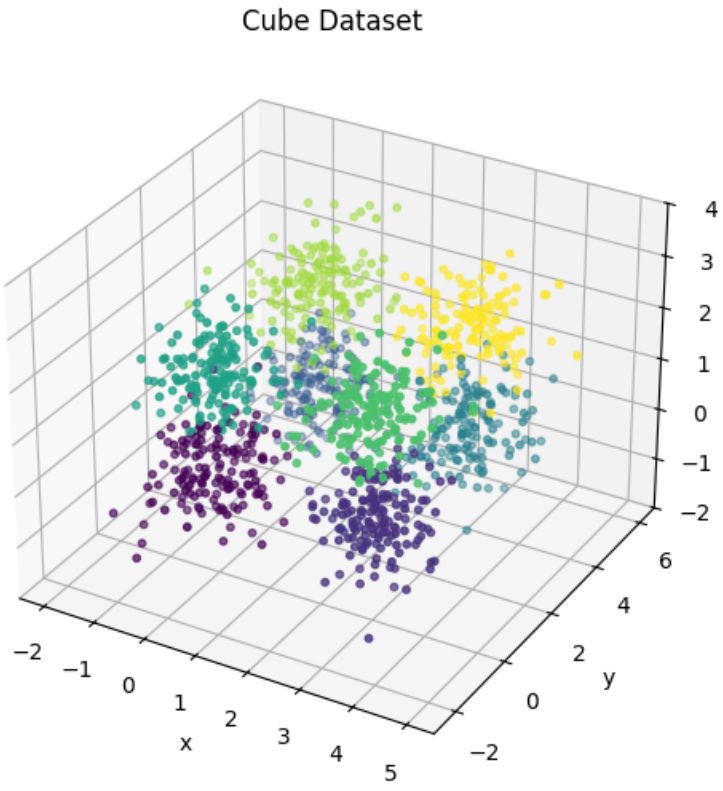
Rys. 3. Wizualizacja sieci z przypisanymi klastrami.

Rezultaty

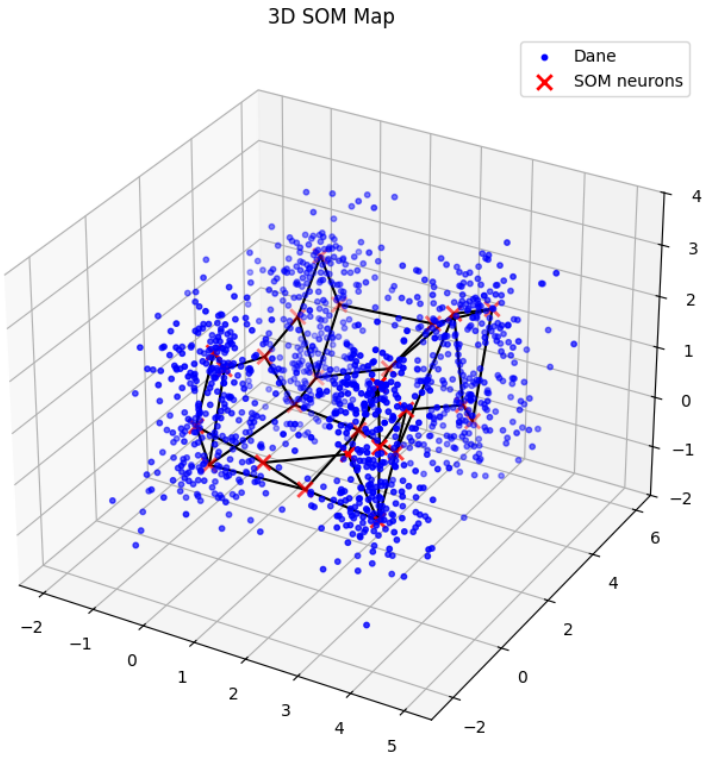
— Rzeczywista liczba klas: 6
— Liczba klas po klasyfikacji: 6
— Dokładność klasyfikacji: 97.83%
— Liczba poprawnych klasyfikacji: 587/600



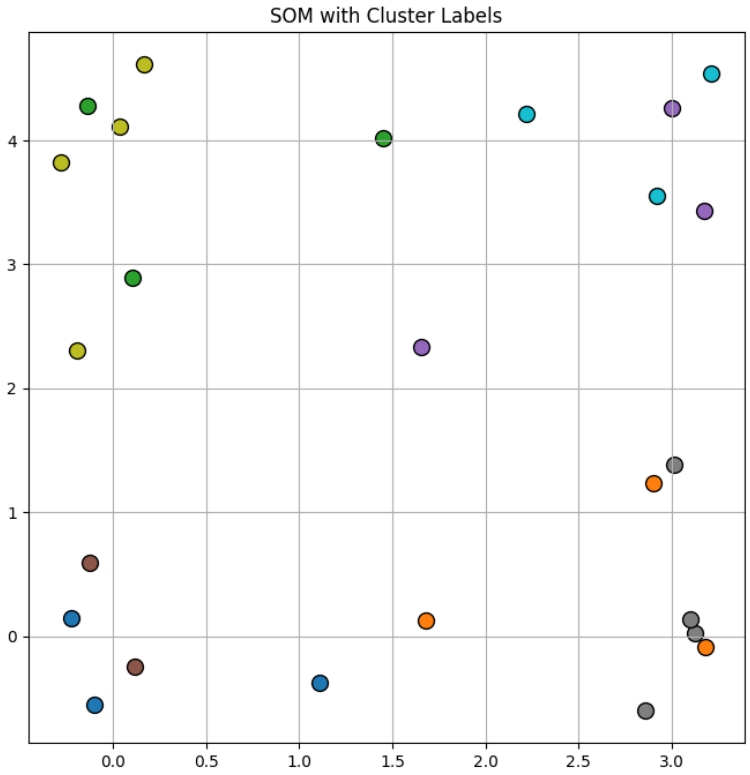
Wyniki dla cube dataset



Rys. 4. Wizualizacja zbioru cube.



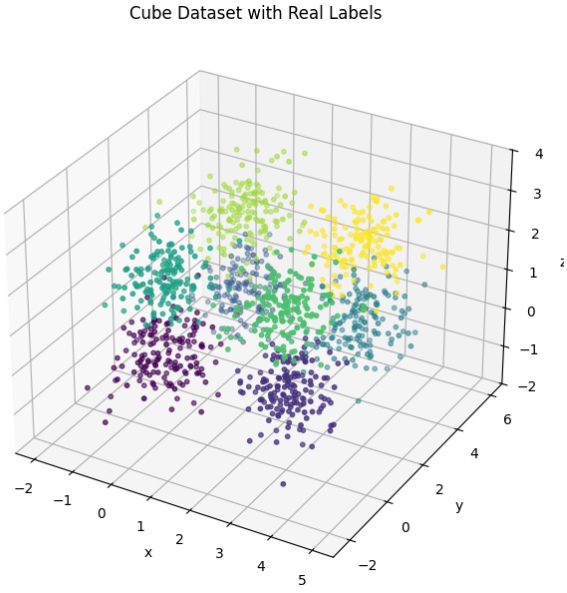
Rys. 5. Wizualizacja sieci Kohonena po trenowaniu (sieć 5x5, 100 epok).

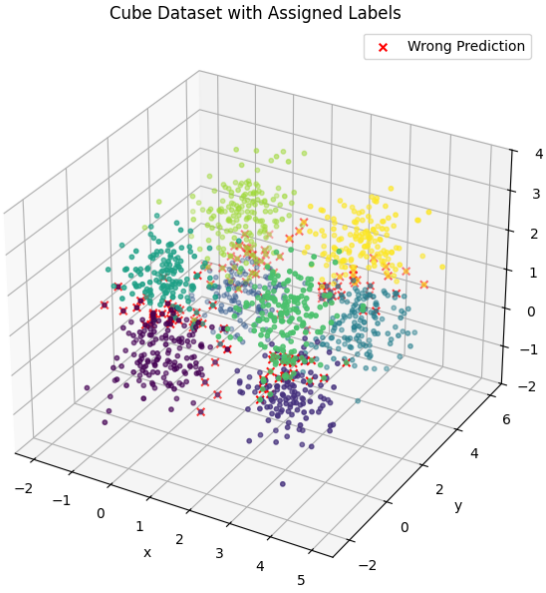


Rys. 6. Wizualizacja sieci z przypisanymi klastrami.

Rezultaty

— Rzeczywista liczba klas: 8
— Liczba klas po klasyfikacji: 8
— Dokładność klasyfikacji: 89.33%
— Liczba poprawnych klasyfikacji: 1072/1200

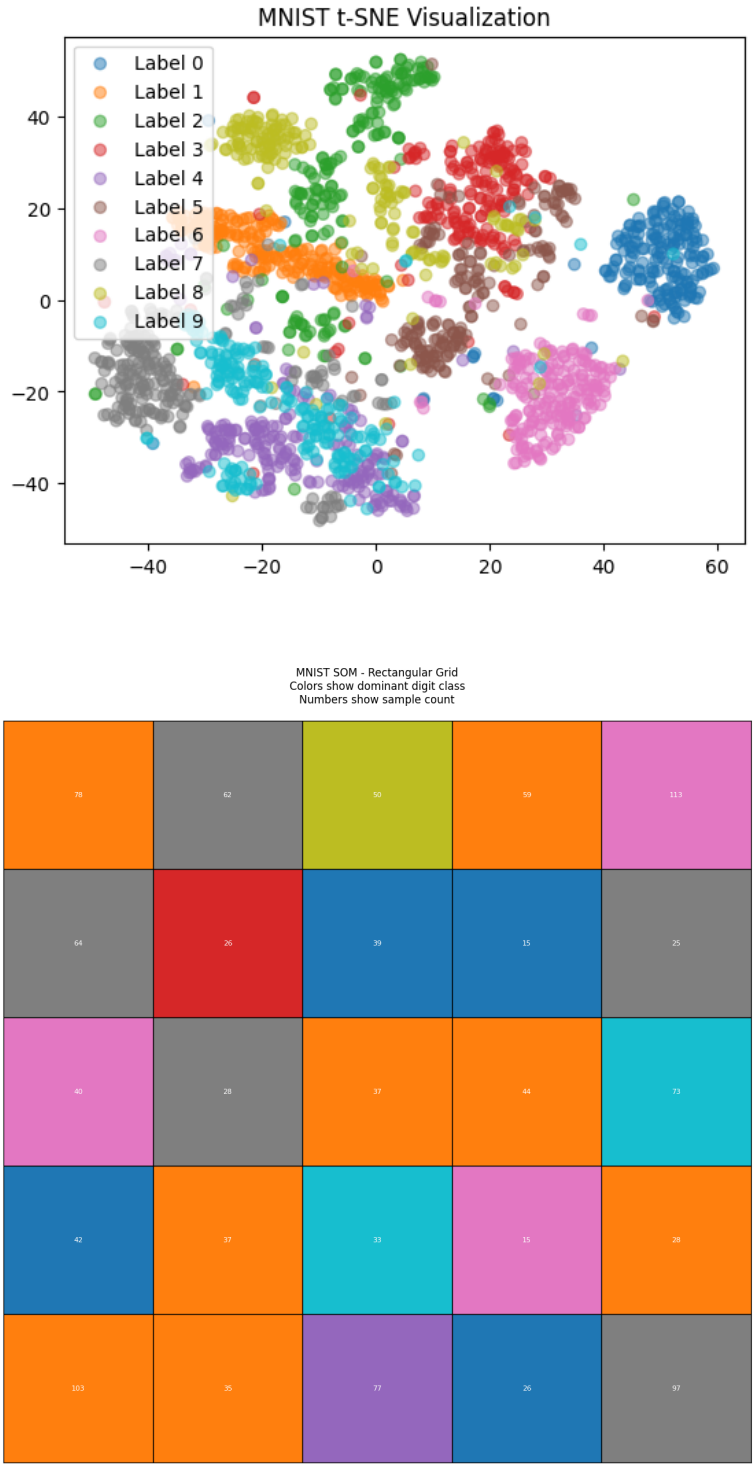




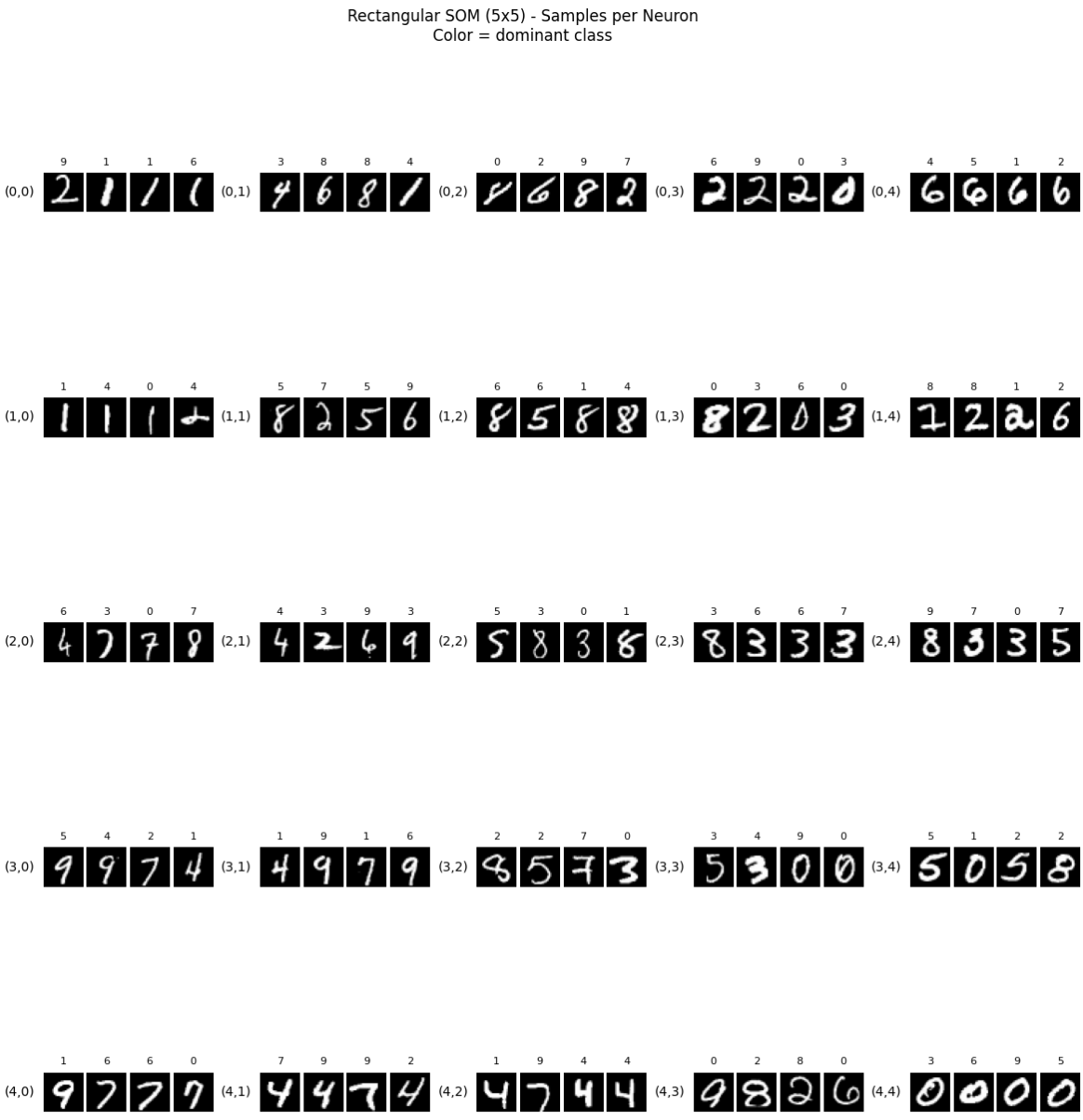
KOH2: Rozszerzenie o topologię heksagonalną

Tym razem wykorzystane zostały zbiory MNIST i Human Activities o dużej wymiarowości

MNIST

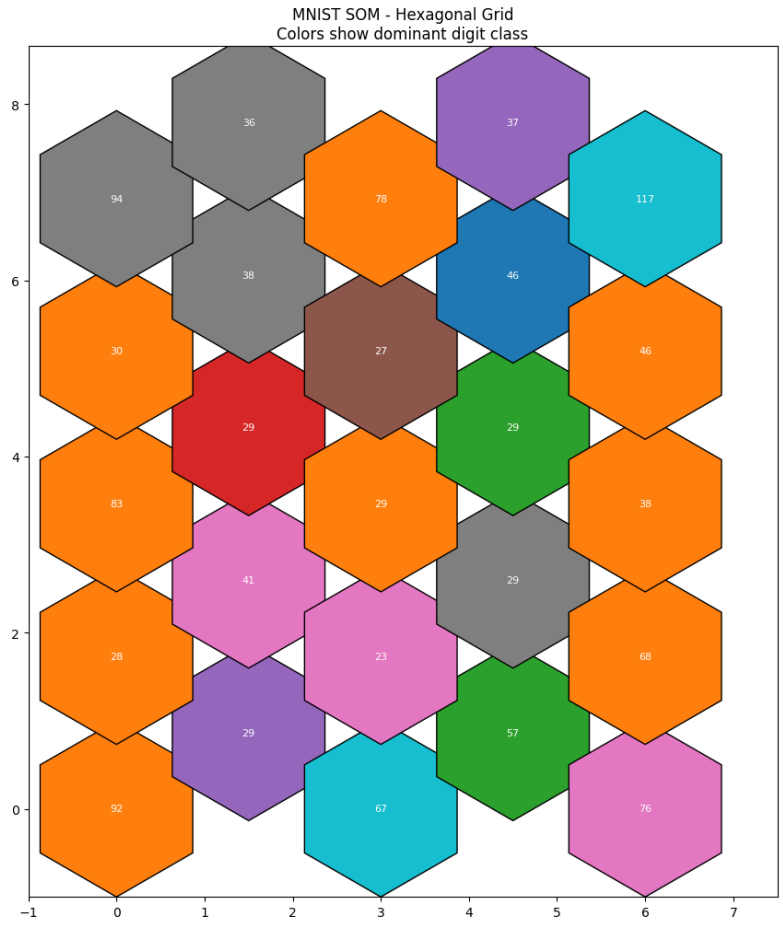


Rys. 7. Wizualizacja sieci (kolor oznacza dominujący klaster)

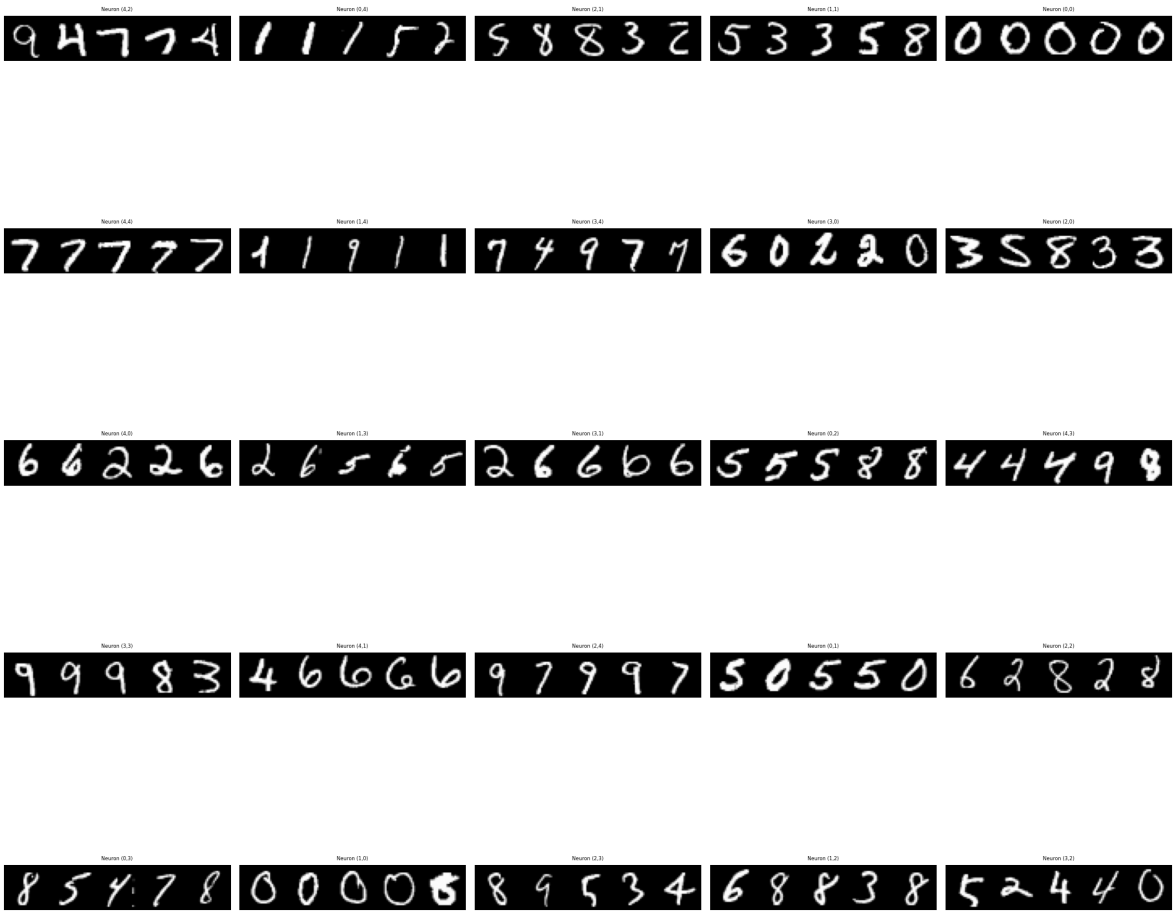


Rys. 8. Przykładowe liczby przyporządkowane do danych neuronów.

Dobre dostrojenie parametrów i połączenie bliskich sobie neuronów we wspólne klastry pozwoliłoby otrzymać zadowalające wyniki i dobrą segmentacje.

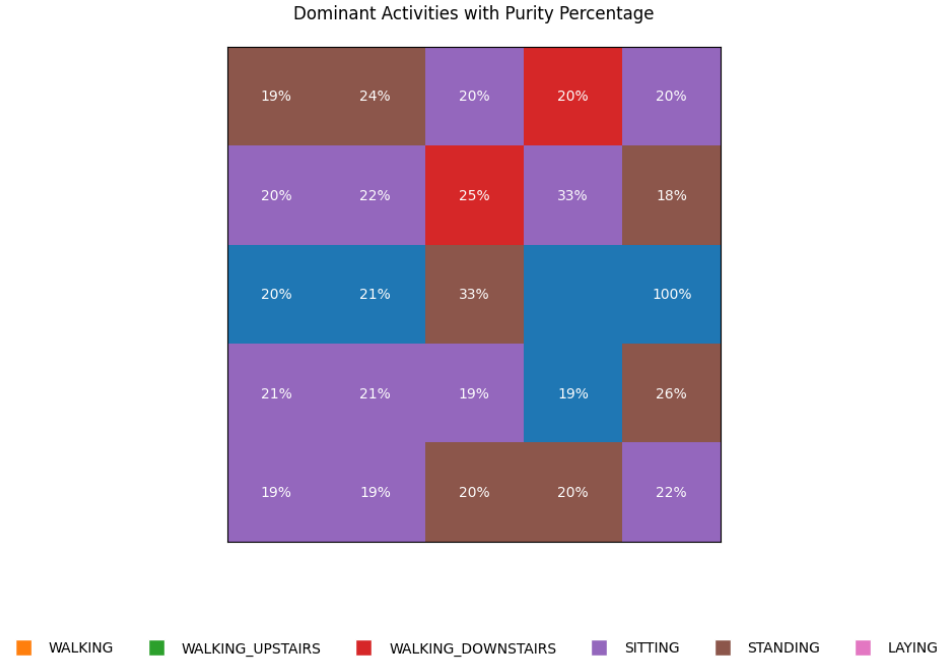


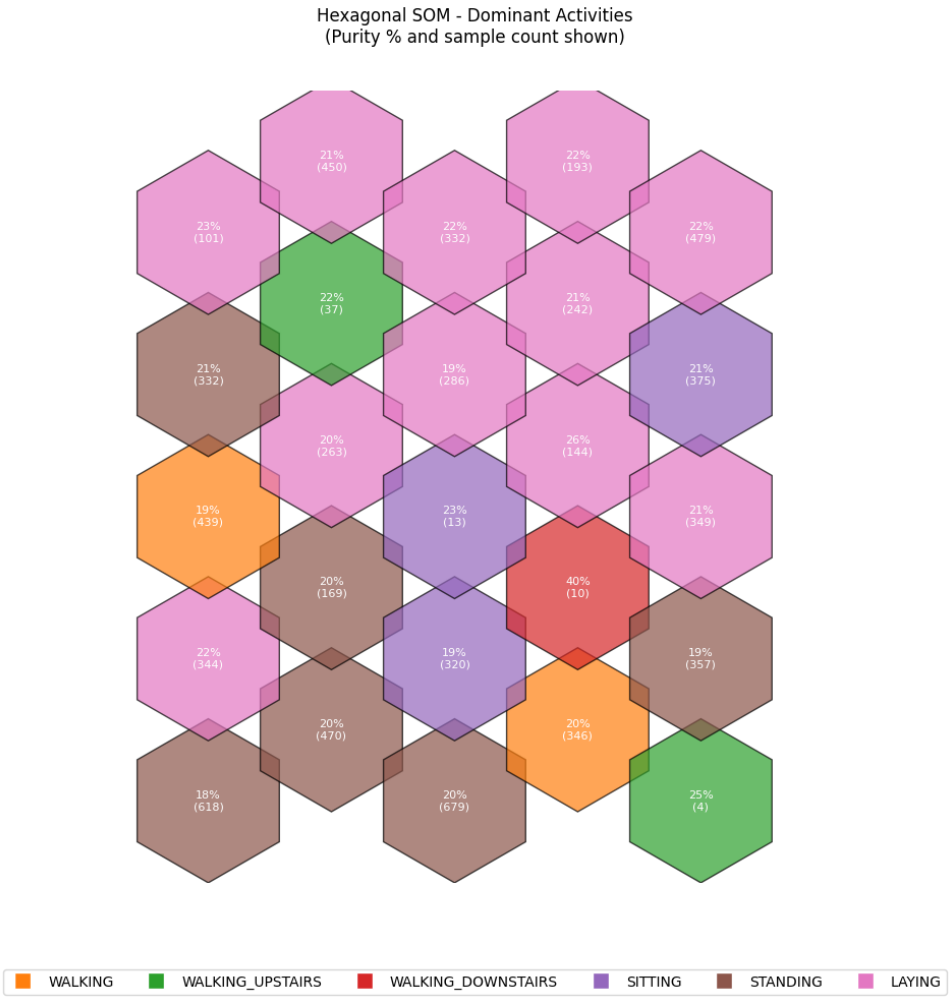
Rys. 9. Wizualizacja sieci (kolor oznacza dominujący klaster)



Rys. 10. Przykładowe liczby przyporządkowane do danych neuronów.

Human Activities





Wyzwania w analizie danych HAR

Główne problemy

— **Nakładanie się cech aktywności statycznych**:
  — Siedzenie vs. stanie wykazują podobne charakterystyki sensoryczne (niska zmienność przyspieszenia)
  — W zbiorze HAR występuje średnia purity tylko 65% dla tych klas vs 82% dla aktywności dynamicznych
  — Przykładowe neurony: (2,3) - 58% sitting, 42% standing

Wizualizacja problemów

```
Neuron (1,2): SITTING (54.3% pure, 127 samples)
Neuron (1,3): STANDING (61.2% pure, 98 samples)
Neuron (2,2): MIXED (34% SITTING, 33% STANDING, 33% LAYING)
```

Przyczyny obserwowanych zjawisk

  — **Fizyczna podobność aktywności**:
    — Współczynnik korelacji cech: 0.78 (sitting vs standing) vs 0.12 (walking vs laying)
  — **Ograniczenia topologiczne**:
    — Sieć 5x5 neuronów dla 6 klas to zbyt mała rozdzielczość
    — Przykład: gdy 3 neurony muszą reprezentować 4 aktywności statyczne
  — **Problem z funkcją sąsiedztwa**:

$$\text{Gaussian} \rightarrow \text{rozmyte granice}, \quad \text{Mexican Hat} \rightarrow \text{puste neurony}$$

Podsumowanie

— Sieć Kohonena skutecznie odwzorowuje strukturę danych — neurony samoorganizują się w taki sposób, że podobne dane trafiają do sąsiednich neuronów.
— W przypadku danych syntetycznych (hexagon, cube) możliwe było dokładne odwzorowanie klastrów. Neurony aktywowały się w rejonach odpowiadających konkretnym wierzchołkom figur.
— W danych MNIST uzyskane klastry SOM częściowo odpowiadają klasom rzeczywistym. Dla niektórych klas (np. cyfr 0, 1, 7) przypisania były wyraźne, natomiast dla innych (np. 4, 9) często występowały rozmycia między klastrami.
— Topologia sześciokątna pozwala na nieco bardziej naturalne odwzorowanie sąsiedztwa — neurony mają sześciu sąsiadów, co zmniejsza efekt "brzegów" siatki i prowadzi do lepszej organizacji przestrzennej danych..

Zastosowania

Sieć Kohonena znajduje zastosowanie w:
— klasteryzacji i eksploracji danych,
— kompresji i redukcji wymiarowości danych,
— wizualizacji danych wysokowymiarowych (np. MNIST),

Ograniczenia

— SOM nie radzi sobie dobrze z bardzo złożonymi zbiorami, gdzie granice między klasami są nieregularne lub dane są silnie zakłócone (zbiór Human Activities).
— Wymaga odpowiedniego doboru parametrów (rozmiar siatki, sąsiedztwo, tempo uczenia), co może być trudne bez wiedzy o danych.
— Sieć nie uczy się etykiet — klasy trzeba analizować dopiero po treningu, co ogranicza możliwości zastosowań nadzorowanych.