

Sieci Kohonena - sprawozdanie

Zuzanna Sienko

Maj 2025

1 Wstęp

Sprawozdanie zawiera opis oraz wyniki eksperymentów przeprowadzonych w ramach zajęć z Metod Inteligencji Obliczeniowej w Analizie Danych z bloku tematycznego Sieci Kohonena (KOH). Każda z sekcji obejmuje analizę poszczególnych podzadań związanych z tym tematem oraz przedstawia eksperymenty realizowane w ramach wybranego zagadnienia.

2 Implementacja tematów

2.1 KOH1 - Podstawowa sieć Kohonena

2.1.1 Opis zadania

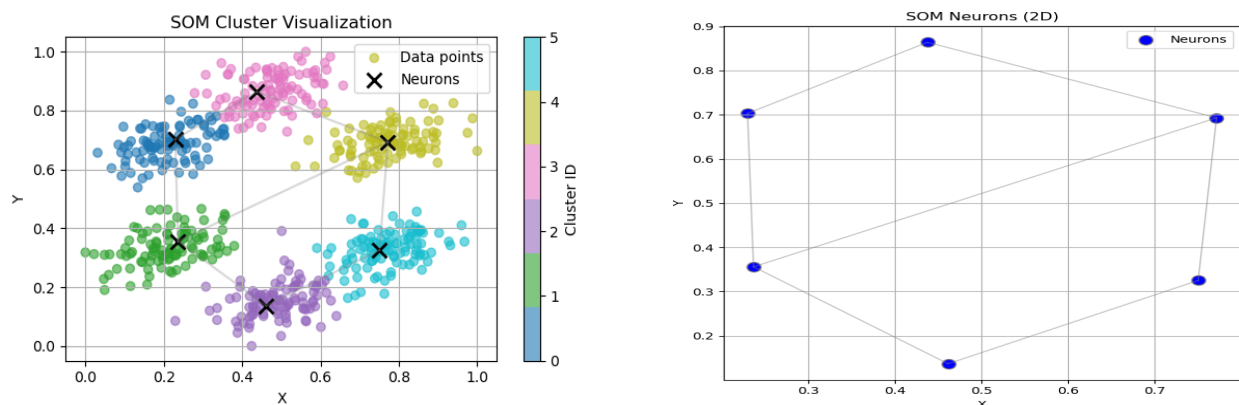
Zadanie polegało na zaimplementowaniu sieci Kohonena na siatce prostokątnej o wymiarach $M \times N$, która działa dla zbioru wektorów (założenie - wszystkie wektory tej samej długości). Wykorzystane funkcje sąsiedztwa to funkcja gaussowska oraz minus druga pochodną funkcji gaussowskiej (mexican hat). Uwzględniona została możliwość zmiany szerokości funkcji sąsiedztwa (parametr s). Wygaszenie współczynnika sąsiedztwa zaimplementowane zgodnie z poleceniem w zadaniu.

Wizualizacje sieci uwzględniały rozkłady siatki neuronów po wytrenowaniu sieci i porównanie ich położenia względem danych z uwzględnieniem Best Matching Unit dla każdej obserwacji, U-Matrix - pokazuje odległości między sąsiednimi neuronami w przestrzeni wag, co pomaga identyfikować skupiska w danych oraz granice klastrów (niskie wartości - podobne neurony, regiony skupisk, wysokie wartości - duże różnice, granice między klastrami).

W procesie dopasowania sieci do danych monitorowano również metryki takie jak Quantization Error (średnia odległość neuronów od BMU), Silhouette Score (jak klasy są wewnętrznie spójne oraz odseparowane od innych klas, zakres $[-1,1]$, im wyższa wartość metryki - lepsza separacja klastrów, nie nakładają się na siebie i są dobrze odseparowane) oraz Davis-Bouldin Index (mierzy stosunek spójności wewnątrz klastra do separacji między klastrami, docelowa wartość 0).

2.1.2 Eksperymenty

Na początku dla zbioru danych hexagon przetestowano różne rozmiary siatek, aby zobaczyć jak neurony dopasowują się do danych i jakie są ich pozycje.



(a) Dopasowanie neuronów do danych hexagonal po dopasowaniu sieci, kolorami zaznaczone dane, który zostały dopasowane do tego samego BMU, siatka 2x3.

(b) Rozkład neuronów po dopasowaniu sieci do danych, siatka 2x3.

Rysunek 1: Porównanie wyników dopasowania sieci Kohonenea dla siatki 2x3.

Dla każdej z wytrenowanych sieci Kohonena generowano raporty opisujące właściwości poszczególnych neuronów, w tym: licznosc przypisanych próbek, rozkład klas w ich obrębie, klasę dominującą oraz czystość przyporządkowania. Poniżej podsumowania dla sieci z Rysunku 1.

Tabela 1: Rozkład klas i dominujące klasy w neuronach SOM

| Neuron | Liczba próbek | Rozkład klas | Dominująca klasa | Czystość (%) | Klasy |
|--------|---------------|--------------------------|------------------|--------------|-----------|
| (0,0) | 97 | 0:1 (1.0%), 5:96 (99.0%) | 5 | 99.0 | [0, 5] |
| (0,1) | 98 | 4:98 (100.0%) | 4 | 100.0 | [4] |
| (0,2) | 100 | 2:1, 3:97, 4:2 | 3 | 97.0 | [2, 4, 3] |
| (1,0) | 102 | 0:98, 5:4 | 0 | 96.1 | [0, 5] |
| (1,1) | 100 | 0:1, 1:99 | 1 | 99.0 | [1, 0] |
| (1,2) | 103 | 1:1, 2:99, 3:3 | 2 | 96.1 | [1, 2, 3] |

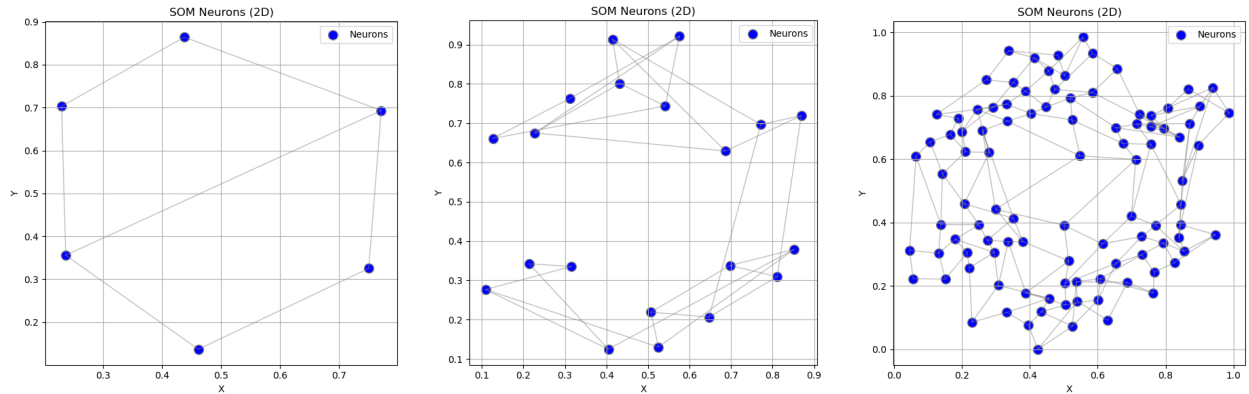
Tabela 2: Podsumowanie przypisania klas do neuronów

| Klasa | Neurony reprezentujące daną klasę |
|-------|---|
| 0 | 3 neurony \rightarrow (0,0), (1,0), (1,1) |
| 1 | 2 neurony \rightarrow (1,1), (1,2) |
| 2 | 2 neurony \rightarrow (0,2), (1,2) |
| 3 | 2 neurony \rightarrow (0,2), (1,2) |
| 4 | 2 neurony \rightarrow (0,1), (0,2) |
| 5 | 2 neurony \rightarrow (0,0), (1,0) |

Zastosowanie raportów dotyczących neuronów w sieci Kohonena pozwoliło odpowiedzieć na kluczowe pytania dotyczące struktury klastrow w danych. Dzięki informacjom o dominujących klasach, czystości neuronów oraz licznosci klas przypisanych do poszczególnych jednostek, możliwa była analiza zgodności odwzorowania z rzeczywistą liczbą klas, a także ocena jednorodności etykiet w obrębie neuronów.

Na tej podstawie każdy neuron został przypisany do jednej dominującej klasy, a wszystkim danym, dla których był on najlepszą jednostką dopasującą (BMU), nadawano etykietę odpowiadającą tej klasie.

Przeprowadzono eksperymenty z różnymi rozmiarami i kształtami siatek. Kluczowym czynnikiem wpływającym na jakość wyników była liczba neuronów, a nie kształt siatki. Wyniki pokazują, że zmiana kształtu siatki (np. z prostokątnej na kwadratową) nie miała istotnego wpływu na monitorowane metryki klasteryzacji. Zwiększenie liczby neuronów w sieci prowadziło do stopniowej poprawy wyników, szczególnie w kontekście obniżenia błędu kwantyzacji oraz poprawy wskaźników oceny jakości klas (Silhouette Score i Davies-Bouldin index).



(a) Sieć Kohonena z siatką 2x3 trenowana na zbiorze hexagonal.

(b) Sieć Kohonena z siatką 10x2 trenowana na zbiorze hexagonal.

(c) Sieć Kohonena z siatką 10x10 trenowana na zbiorze hexagonal.

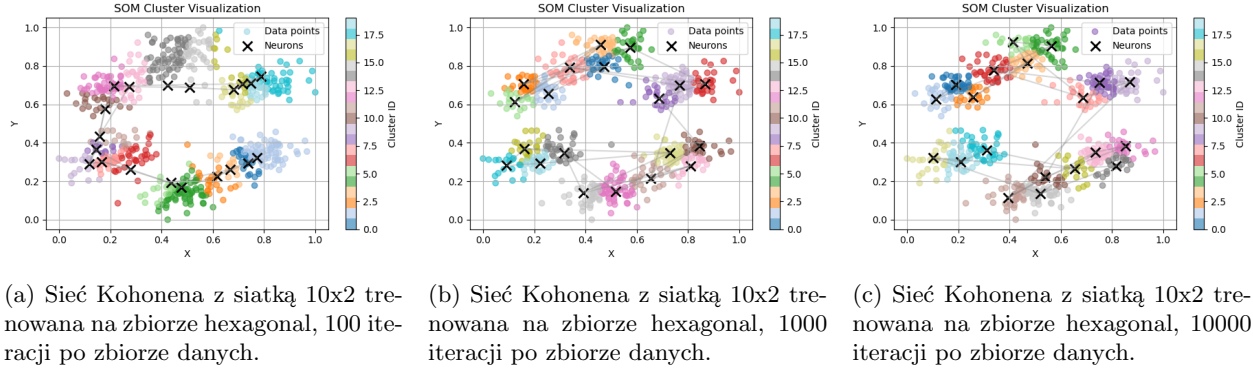
Rysunek 2: Porównanie rozkładu neuronów stosując różne rozmiary siatek dla zbioru hexagonal.

Na przedstawionych rysunkach można zaobserwować, że dla każdej konfiguracji siatki neurony przyjęły charakterystyczny układ sześciokątny, co wskazuje na efektywne odwzorowanie topologii danych przez sieć Kohonena.

Tabela 3: Wpływ liczby neuronów na jakość klasteryzacji w sieci Kohonena

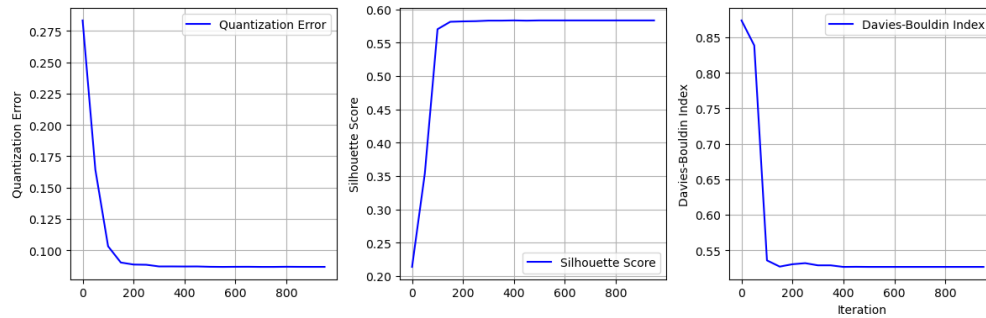
| M | N | Liczba neuronów | Liczba klastrów | Błąd kwantyzacji | Silhouette | DB Index |
|----|----|-----------------|-----------------|------------------|------------|---------------|
| 2 | 2 | 4 | 4 | 0.1492 | 0.4753 | 0.6779 |
| 5 | 5 | 25 | 25 | 0.0486 | 0.3220 | 0.9013 |
| 10 | 10 | 100 | 95 | 0.0226 | 0.3559 | 0.6900 |
| 25 | 5 | 125 | 124 | 0.0187 | 0.3489 | 0.6916 |
| 25 | 25 | 625 | 448 | 0.0027 | 0.2134 | 0.2746 |

Eksperymenty z różną liczbą iteracji pokazały, że im dłużej trwa proces uczenia, tym lepiej sieć Kohonena dopasowuje się do danych. Przy zastosowaniu funkcji sąsiedztwa gaussowskiej zauważono, że już po 100 iteracjach neurony częściowo odwzorowują strukturę danych zbioru hexagonal, jednak dopiero przy większej liczbie iteracji dopasowanie staje się dokładne i stabilne. Oczywiście liczba potrzebnych iteracji zależy również od wybranego współczynnika learning rate decay, kontrolującego tempo uczenia.

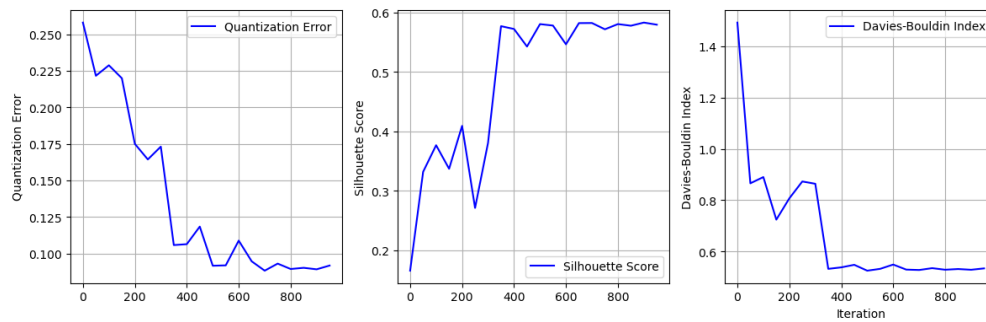


Rysunek 3: Porównanie rozkładu neuronów stosując różne liczby iteracji dla zbioru hexagonal.

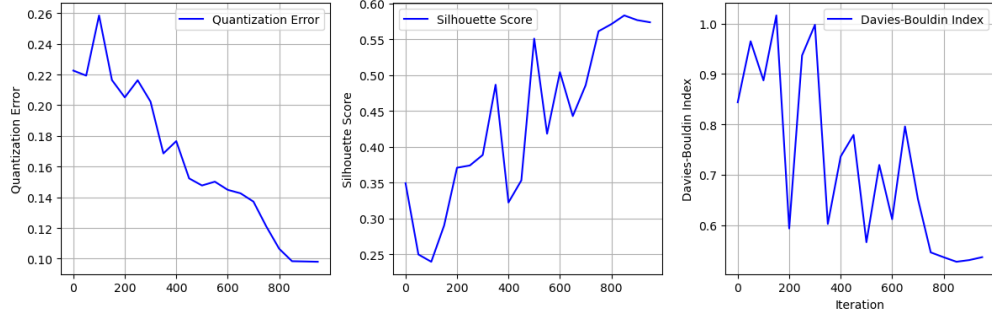
Przeprowadzono eksperymenty z różnymi wartościami parametru λ , który kontroluje stopniowe zmniejszanie się współczynnika uczenia (learning rate decay) w trakcie procesu uczenia sieci Kohonena. Przeprowadzono eksperymenty z wartościami $\lambda = [1, 10, 100, 500, 1000, 2000]$, a wyniki zostały zwizualizowane za pomocą wykresów, które przedstawiają wpływ tych wartości na model po przeprowadzeniu 1000 iteracji.



Rysunek 4: Dopasowanie sieci Kohonena z wartością lambda 100.



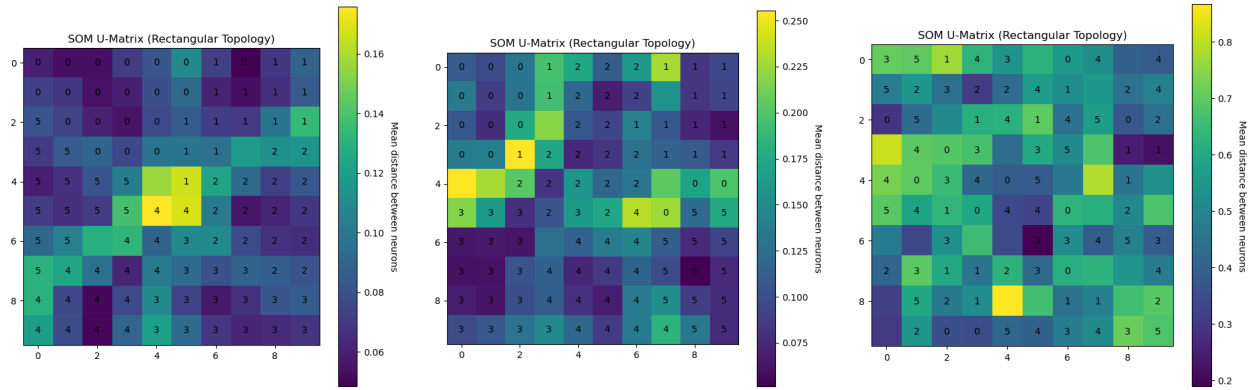
Rysunek 5: Dopasowanie sieci Kohonena z wartością lambda 500.



Rysunek 6: Dopasowanie sieci Kohonena z wartością lambda 1000.

Odpowiedni dobór wartości parametru regulującego współczynnik uczenia jest istotny, ponieważ wpływa na stabilność i efektywność procesu uczenia. Zbyt niska wartość może prowadzić do wolnej zbieżności, natomiast zbyt wysoka – do niestabilności modelu i gorszej jakości mapowania.

Przetestowano również różne wartości dla zmiany szerokości sąsiedztwa (parametr s) - kontrolującego ile neuronów jest uaktualnianych w sąsiedztwie BMU i przeanalizowano wpływ parametru na podstawie U-matrix. Wyraźne jasne linie lub obszary sugerują dobrze zdefiniowane granice między klasami. Duże ciemne regiony wskazują na gęste skupiska neuronów reprezentujących jedną klasę.



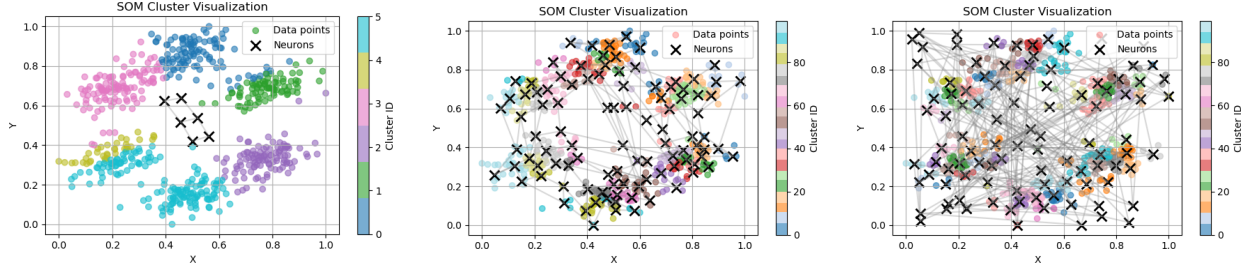
(a) U-matrix dla sieci Kohonena wytrenowanej z parametrem $s=0.1$.

(b) U-matrix dla sieci Kohonena wytrenowanej z parametrem $s=1.0$.

(c) U-matrix dla sieci Kohonena wytrenowanej z parametrem $s=10.0$.

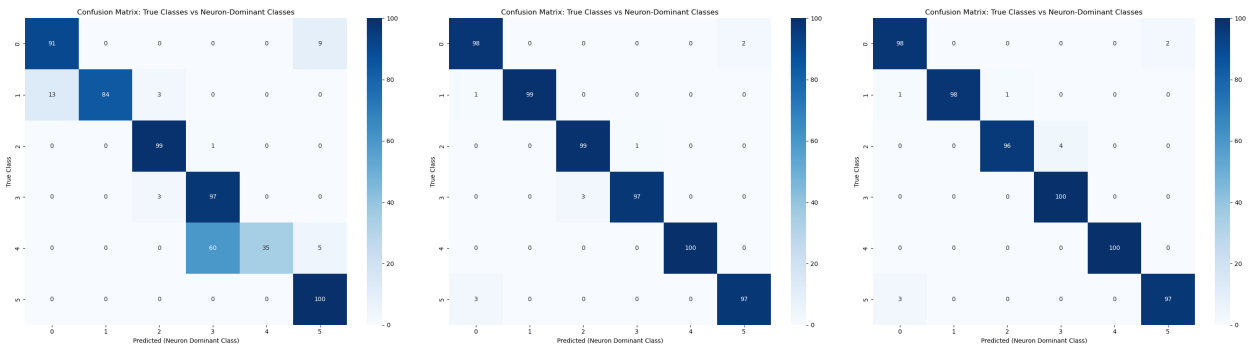
Rysunek 7: Porównanie U-matrix w zależności od wartości parametru skalującego wielkość sąsiedztwa.

Przy bardzo małym $s = 0.1$ U-matrix pokazuje jeden jasny punkt, a reszta jest raczej ciemna i jednolita – sieć wychwytuje lokalne różnice, ale nie oddaje szerszej struktury. Widać, że dominujące klasy w neuronach są ułożone obok siebie, tworząc skupiska. Dla $s = 1.0$ obraz staje się bardziej zróżnicowany – lepiej widoczne są granice między klasami i układ grup danych, a klasy nadal pozostają pogrupowane w sąsiednich neuronach. Z kolei przy $s = 10$ całość wygląda równomiernie i mało kontrastowo – dominujące klasy są rozrzucone bardziej losowo.



(a) Wizualizacja dla sieci Kohonena wytrenowanej z parametrem $s=0.1$. (b) Wizualizacja dla sieci Kohonena wytrenowanej z parametrem $s=1.0$. (c) Wizualizacja dla sieci Kohonena wytrenowanej z parametrem $s=10.0$.

Rysunek 8: Porównanie wizualizacji klasteryzacji sieci w zależności od wartości parametru skalującego wielkość sąsiedztwa.

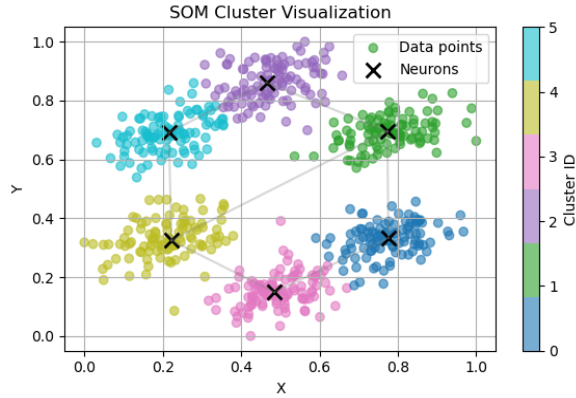


(a) Confusion matrix dla sieci Kohonena wytrenowanej z parametrem $s=0.1$. (b) Confusion matrix dla sieci Kohonena wytrenowanej z parametrem $s=1.0$. (c) Confusion matrix dla sieci Kohonena wytrenowanej z parametrem $s=10.0$.

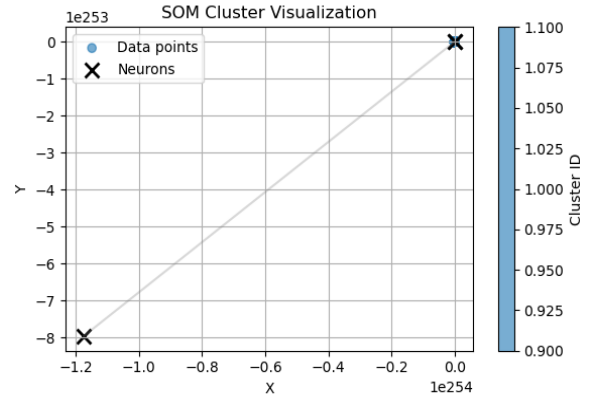
Rysunek 9: Porównanie wyników klasteryzacji sieci w zależności od wartości parametru skalującego wielkość sąsiedztwa.

Jak pokazuje confusion matrix, sieć z $s = 0.1$, uwzględniająca bardzo mały obszar sąsiedztwa, osiągnęła najslabsze wyniki klasyfikacyjne. Najlepszą dokładność uzyskano przy $s = 1.0$, choć $s = 10$ dało wyniki porównywalne.

Na zbiorze hexagonal przeprowadzono eksperymenty z wykorzystaniem funkcji sąsiedztwa typu mexican hat. W przeciwieństwie do funkcji gaussowskiej, dla której dobór parametrów był stosunkowo prosty, eksperymenty wykazały, że mexican hat jest znacznie bardziej wrażliwa na ustawienia parametrów sieci.



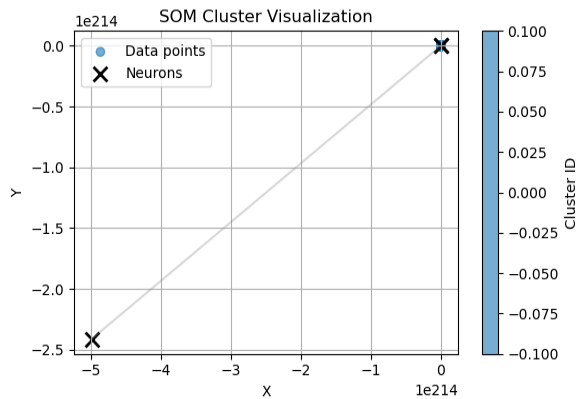
(a) Rozkład neuronów po dopasowaniu sieci do danych z funkcją gaussowską.



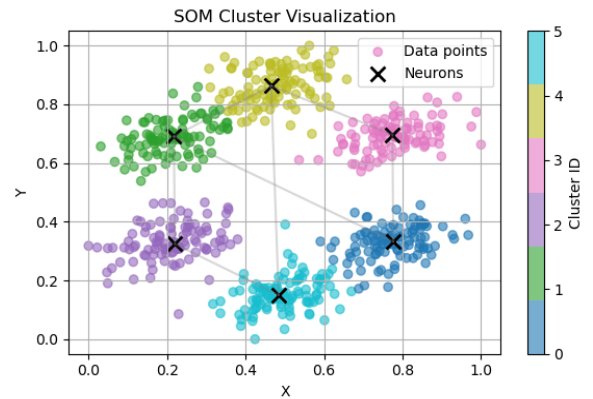
(b) Rozkład neuronów po dopasowaniu sieci do danych z funkcją mexican hat.

Rysunek 10: Porównanie wyników dopasowania sieci Kohonenea dla różnych funkcji sąsiedztwa z takimi samymi parametrami.

Przykładowo eksperymenty pokazały, że parametr skalujący funkcję sąsiedztwa (s) jest jednym z istotnych czynników wpływających na działanie funkcji mexican hat. Zarówno przy bardzo niskiej wartości ($s=0.1$), jak i przy domyślnej ($s=1.0$), sieć przypisywała większość próbek do jednego neuronu, co uniemożliwiało sensowne odwzorowanie struktury danych. Dopiero przy $s=10$ pojawiło się wyraźniejsze grupowanie i lepsze odwzorowanie klas. Podobnie funkcja mexican hat była bardzo wrażliwa na odpowiedni dobór parametru kontrolującego spadek współczynnika uczenia.

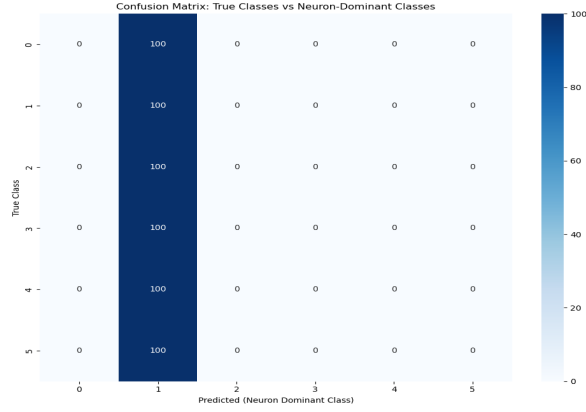


(a) Rozkład neuronów po dopasowaniu sieci do danych z funkcją mexican hat $s=0.1$.

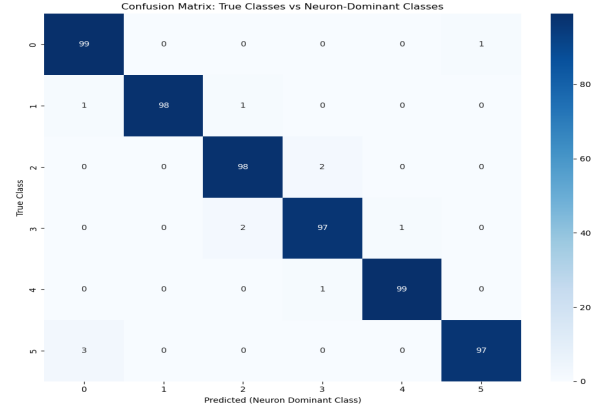


(b) Rozkład neuronów po dopasowaniu sieci do danych z funkcją mexican hat $s=10$.

Rysunek 11: Porównanie wyników dopasowania sieci Kohonenea dla różnych wartości parametru s .



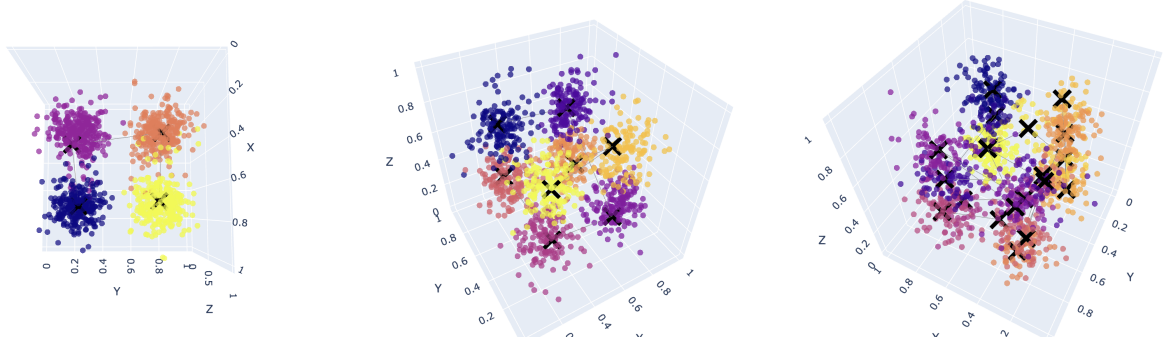
(a) Confusion matrix po dopasowaniu sieci do danych z funkcją mexican hat $s=0.1$.



(b) Confusion matrix po dopasowaniu sieci do danych z funkcją mexican hat $s=10$.

Rysunek 12: Porównanie wyników dopasowania sieci Kohonenea dla różnych wartości parametru s .

Dla zbioru danych cube stosunkowo szybko udało się dobrać odpowiednie parametry sieci Kohonenea, eksperymenty analogiczne do zbioru hexagonal. Widoczne jest dopasowywanie się neuronów do rozkładu danych w przestrzeni cech.

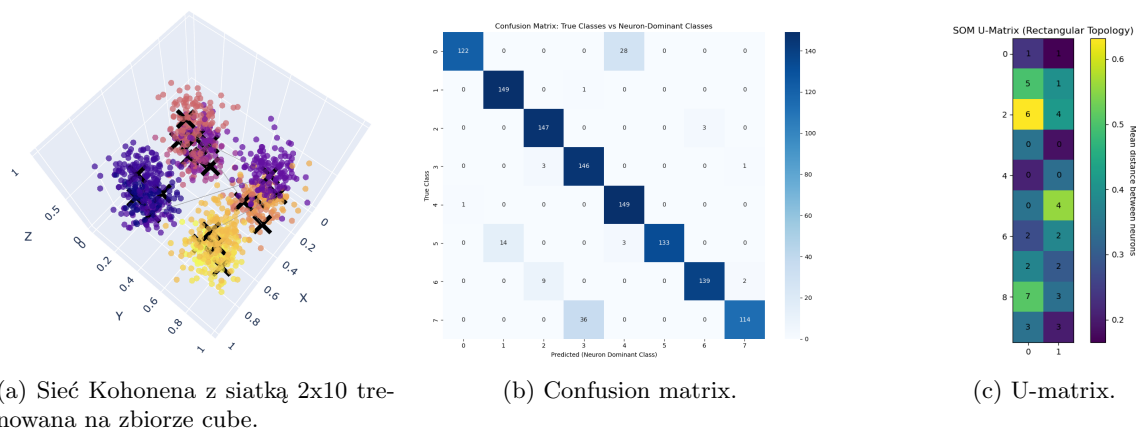


(a) Sieć Kohonenea z siatką 2x2 trenowana na zbiorze cube.

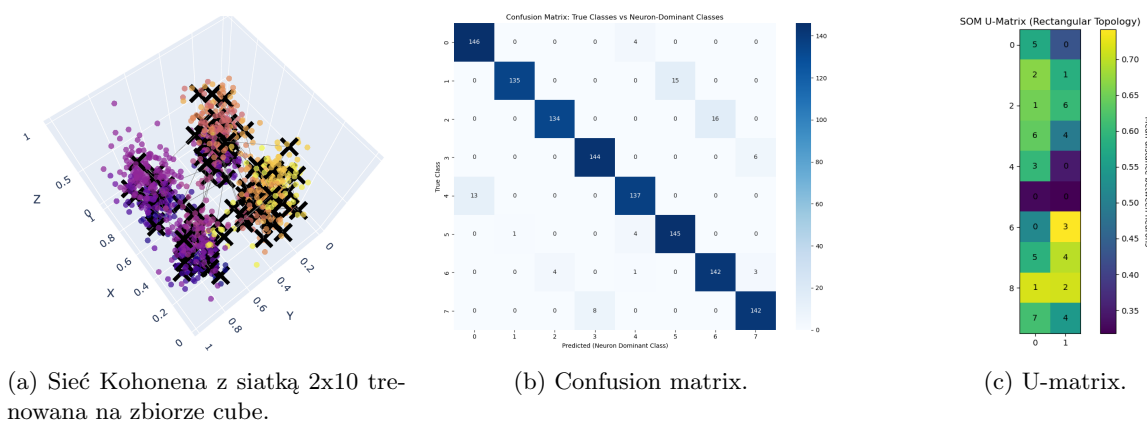
(b) Sieć Kohonenea z siatką 2x4 trenowana na zbiorze cube.

(c) Sieć Kohonenea z siatką 2x19 trenowana na zbiorze cube.

Rysunek 13: Porównanie rozkładu neuronów stosując różne wielkości siatek dla zbioru cube.



Rysunek 14: Dopasowanie sieci Kohonena do zbioru cube z siatką 2x10 i funkcją gaussowską.



Rysunek 15: Dopasowanie sieci Kohonena do zbioru cube z siatką 2x10 i funkcją mexican hat.

Na rysunkach 14 i 15 przedstawiono dopasowanie sieci Kohonena do zbioru danych cube, wytrenowanej na tej samej siatce. W przypadku zastosowania funkcji mexican hat, sieć wykazała lepsze odwzorowanie rzeczywistych klastrów, co jest wyraźnie widoczne w confusion matrix.

2.1.3 Wnioski

Sieć Kohonena skutecznie klasteryzowała dane, a wyniki pokazały, że większa liczba neuronów i iteracji poprawia jakość odwzorowania danych. Wartość parametru współczynnika uczenia (λ) miała istotny wpływ na stabilność procesu, a optymalna szerokość funkcji sąsiedztwa pozwalała uzyskać lepsze wyniki w klasteryzacji. Zwiększenie liczby iteracji umożliwiło dokładniejsze dopasowanie modelu do danych, co przyczyniło się do lepszej jakości klastrów. Wnioski wskazują, że właściwy dobór parametrów, takich jak liczba neuronów, iteracji oraz funkcja sąsiedztwa, jest kluczowy dla uzyskania dokładnych wyników.

2.2 KOH2 - Sieć Kohonena na siatce sześciokątnej.

2.2.1 Opis zadania

Do poprzedniej implementacji została dodana możliwość ustawienia topologii siatki (rectangular - prostokątnej, hexagonal - sześciokątnej). Topologie i funkcje sąsiedztwa (gaussian i mexican hat) zostały przetestowane na zbiorach danych MNIST - zbiór zawierający obrazy ręcznie pisanych liczb 0-9 oraz Human Activity Recognition Using Smartphones.

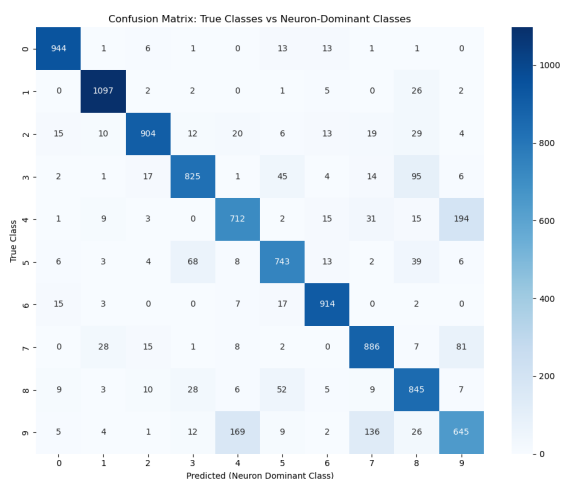
Aby ocenić dokładność klasteryzacji dla zbiorów o wysokiej wymiarowości (MNIST z obrazów 28x28 pikseli przekonwertowany na wektor 784, Human Activity Recognition 561 cech) dla każdego neuronu znajdowano dominującą etykietę ze zbiorów punktów dla których ten neuron był BMU.

2.2.2 Eksperymenty

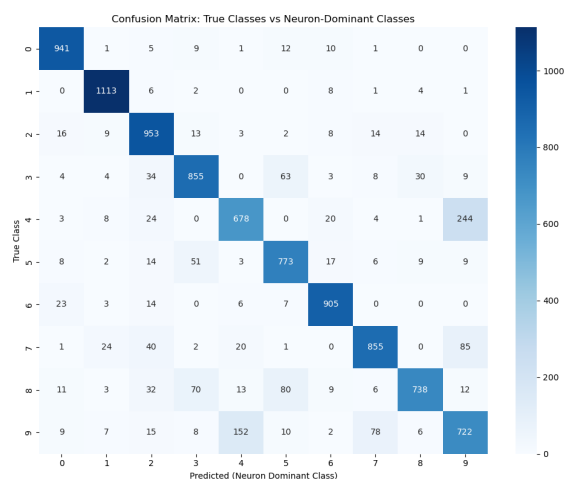
Dla każdego zbioru danych przeprowadzono testy porównawcze obu topologii oraz obu funkcji sąsiedztwa.

Dla zbioru MNIST wylosowano próbkę, na której została dopasowana sieć (ze względu na redukcję czasu obliczeniowego), a następnie przetestowano wyniki dopasowania na pełnym zbiorze testowym. W celu oceny jakości klasteryzacji zastosowano confusion matrix porównując przewidywane klasy z rzeczywistymi etykietami oraz jak wyglądały U-matrix (z zaznaczeniem dominującej klasy dla każdego neuronu). Monitorowano również metryki klasteryzacyjne opisane w poprzednim zadaniu.

Przy prostokątnej siatce 10x10 i zastosowaniu funkcji gaussowskiej jako funkcji sąsiedztwa otrzymano wyniki:



Rysunek 16: Confusion matrix dla siatki prostokątnej (dane testowe).

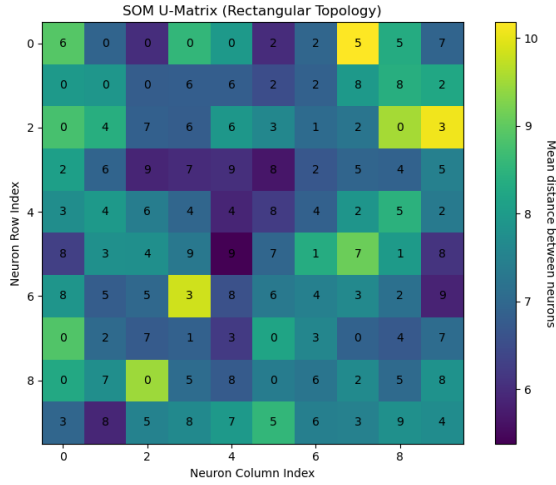


Rysunek 17: Confusion matrix dla siatki sześciokątnej (dane testowe).

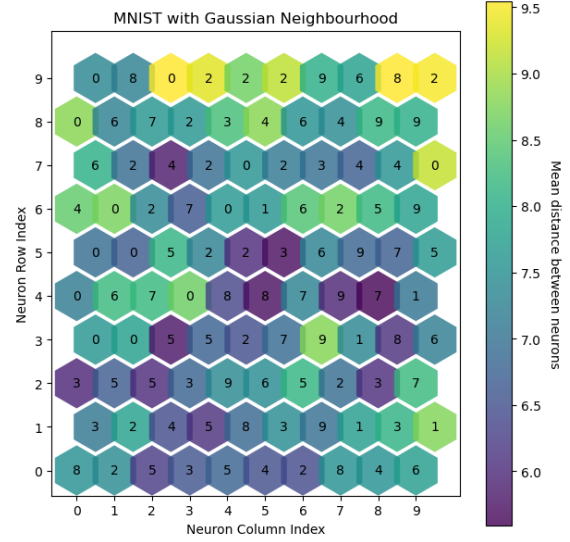
W obu przypadkach, niezależnie od zastosowanej topologii, wszystkie neurony wykazywały wysoką czystość, rozumianą jako wysoki odsetek próbek należących do dominującej klasy przypisanych do danego neuronu na podstawie BMU (w większości przypadków 90%).

Prostokątna siatka (na zbiorze testowym) lepiej radzi sobie z parą 4/9, ale słabiej dla 7/9 i 3/8.

Sześciokątna siatka (na zbiorze testowym) zapewnia lepsze rozróżnianie 3/8 i 7/9, ale wykazuje mniejszą dokładność klasteryzacji dla 4/9.



Rysunek 18: U-matrix dla siatki prostokątnej



Rysunek 19: U-matrix dla siatki sześciokątnej

Na podstawie U-matrixów (Rysunek 18, 19) prostokątna topologia w U-matrix ma wyraźniejsze granice między klasami 3/8 i 4/9. Natomiast siatka sześciokątna pokazuje wyższy kontrast między neuronami dla klas 7/9, co sugeruje skuteczniejsze ich odseparowanie w tej topologii.

Porównano również etykiety dla których jedna siatka radzi sobie lepiej od drgiej. Przykładowo dla pary etykiet 3 i 8, dla której siatka sześciokątna radziła sobie lepiej od prostokątnej.

Tabela 4: Porównanie neuronów z dominującymi klasami 3 i 8 w dwóch topologiach SOM

| Parametr | Klasa 3 | | Klasa 8 | |
|-------------------------------|-------------|--------------|-------------|--------------|
| | Prostokątna | Sześciokątna | Prostokątna | Sześciokątna |
| Liczba neuronów | 12 | 10 | 9 | 9 |
| Udział klasy dominującej | 77.63% | 84.65% | 88.93% | 90.12% |
| Najczęstsza klasa zakłócająca | 8 (86) | 8 (70) | 5 (34) | 3 (38) |
| Udział klasy zakłócającej | 7.48% | 6.93% | 3.84% | 4.58% |
| Pozostałe klasy (łącznie) | 14.89% | 8.42% | 7.23% | 5.30% |

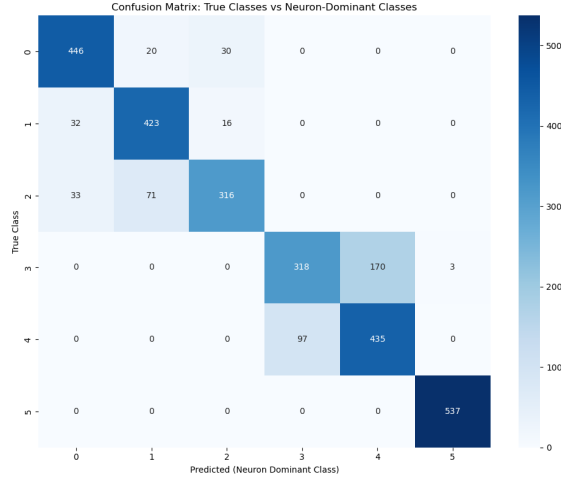
Monitorowane metryki podczas treningu nie różniły się znacząco w zależności od wybranej topologii siatki i osiągnęły zbliżone wyniki.

Tabela 5: Porównanie metryk dla topologii sześciokątnej i prostokątnej

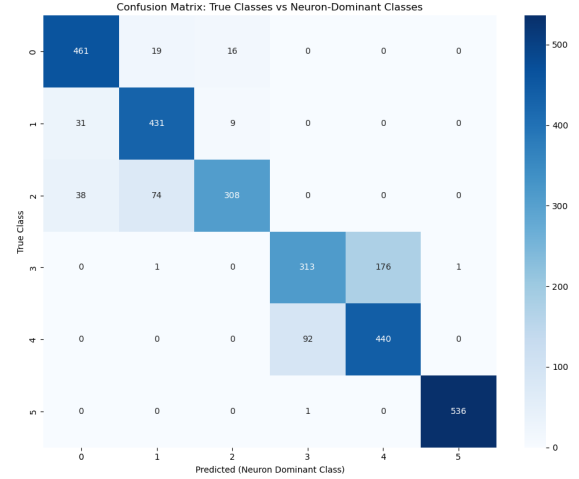
| Metryka | Siatka Sześciokątna | Siatka Prostokątna |
|----------------------|---------------------|--------------------|
| Quantization Error | 5.1558 | 5.1661 |
| Silhouette Score | 0.0597 | 0.0616 |
| Davies-Bouldin Index | 2.5616 | 2.5920 |

Dla funkcji sąsiedztwa mexican hat nie udało się dobrać parametrów, które skutecznie tworzyłyby klastry – niezależnie od zastosowanej topologii, aktywne pozostały jedynie 4 neurony, a reszta siatki pozostała pusta.

Dla zbioru Human Activity Recognition zastosowano takie samo podejście co do zbioru MNIST.

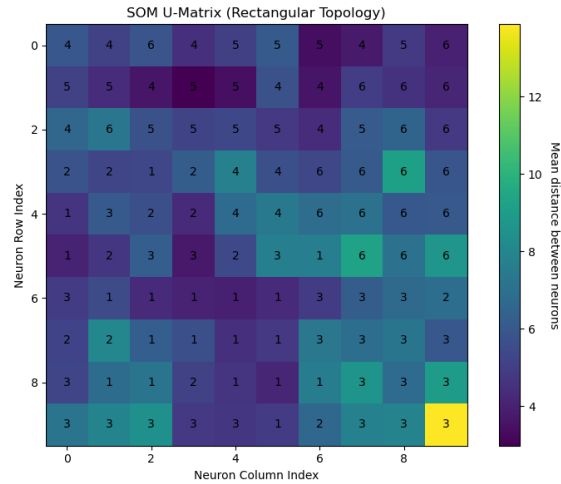


Rysunek 20: Confusion matrix dla siatki prostokątnej (dane testowe).

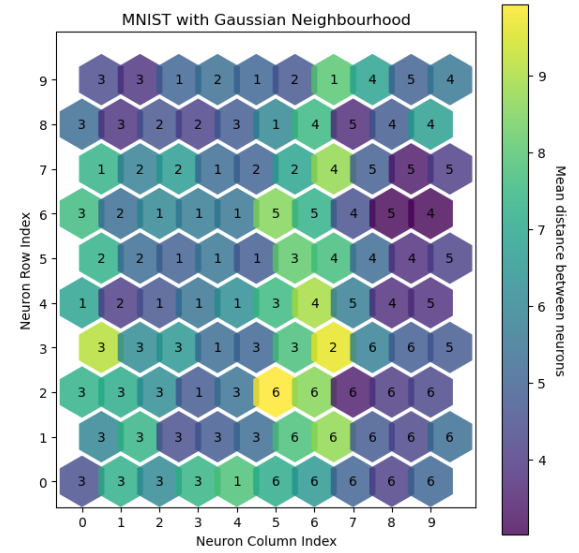


Rysunek 21: Confusion matrix dla siatki sześciokątnej (dane testowe).

Otrzymano bardzo zbliżone wyniki dla obu zastosowanych topologii.



Rysunek 22: U-matrix dla siatki prostokątnej



Rysunek 23: U-matrix dla siatki sześciokątnej

Analiza U-matrix wskazuje, że w topologii prostokątnej granice między klastrami są mniej wyraźne, co prowadzi do większych, mniej rozróżnialnych skupisk neuronów. W topologii sześciokątnej granice między klastrami są bardziej wyraźne, co świadczy o lepszym rozdzieleniu grup danych. Ponadto, średnie odległości między neuronami są mniejsze w siatce sześciokątnej (zakres 4-9) niż w prostokątnej (zakres 4-12), co sugeruje bardziej zwarte rozmieszczenie neuronów w tej topologii.

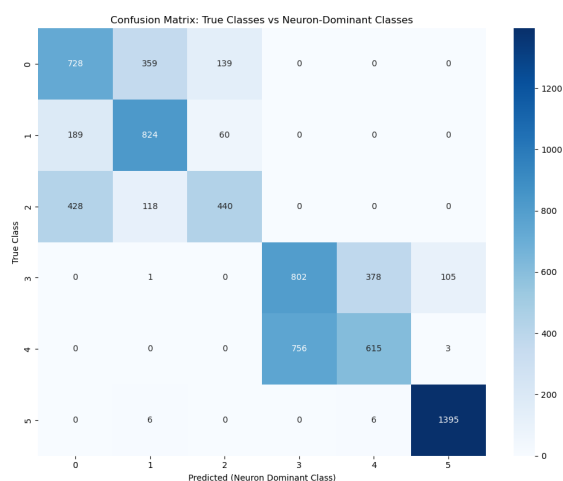
Tabela 6: Porównanie metryk dla topologii sześciokątnej i prostokątnej

| Metryka | Siatka Sześciokątna | Siatka Prostokątna |
|----------------------|---------------------|--------------------|
| Quantization Error | 3.2973 | 3.2886 |
| Silhouette Score | 0.0730 | 0.0781 |
| Davies-Bouldin Index | 2.3637 | 2.2867 |

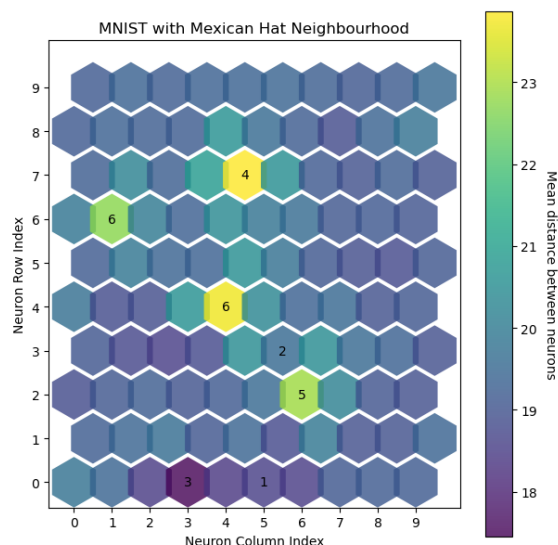
Pomimo że analiza U-matrix wskazuje na bardziej wyraźne granice między klastrami w topologii sześciokątnej — co może sugerować lepsze rozdzielanie przestrzenne neuronów — to metryki ilościowe (Silhouette Score, Davies-Bouldin Index) są nieco korzystniejsze dla topologii prostokątnej. Oznacza to, że w ujęciu statystycznym klastry uzyskane w siatce prostokątnej charakteryzują się większą wewnętrzną spójnością i lepszym rozdzieleniem względem siebie.

Warto jednak zauważyć, że U-matrix odzwierciedla lokalną strukturę mapy, podczas gdy wspomniane metryki oceniają jakość klasteryzacji z punktu widzenia całego zbioru danych.

Dla funkcji mexican hat nie uzyskano zadowalających wyników - większość neuronów pozostała pusta, a czystość neuronów nie była na wysokim poziomie.



Rysunek 24: Confusion matrix dla siatki sześciokątnej z funkcją mexican hat.



Rysunek 25: U-matrix dla siatki sześciokątnej z funkcją mexican hat.

Jak widać zastosowanie funkcji mexican hat prowadzi do ograniczonej aktywacji neuronów i częściowego rozdzielania klas (Rysunek 25). Wiele neuronów grupuje jednocześnie klasy 1, 2 i 3 lub 4, 5 i 6, co skutkuje niską czystością (ok. 54–69%) i błędami w przypisaniu do odpowiednich klastrów (Rysunek 24). Tylko pojedyncze neurony, jak (4,4), osiągają wysoką separację (95.8%). Mapa jest słabo rozproszona – większość siatki pozostaje nieaktywna.

2.2.3 Wnioski

Topologia siatki (prostokątna vs sześciokątna) wpływa głównie na lokalne właściwości mapy, takie jak widoczność granic między klastrami czy stopień rozproszenia neuronów. Sześciokątna topologia może zapewniać bardziej gładkie i ciągłe przejścia między neuronami dzięki większej liczbie sąsiadów, co ułatwia odseparowanie pewnych klas w przestrzeni mapy. Jednak w ujęciu statystycznym, analizując metryki klasteryzacji (Quantization Error, Silhouette Score, Davies-Bouldin Index), różnice są niewielkie, co wskazuje, że wybór topologii ma ograniczony wpływ na ogólną jakość grupowania danych.

Literatura

- [1] J. M. Wandeto and B. Dresp-Langley,
The quantization error in a Self-Organizing Map as a contrast and colour specific indicator of single-pixel change in large random patterns,
Dostępne: <https://arxiv.org/pdf/2011.03970>
- [2] P. J. Rousseeuw,
Silhouettes: a graphical aid to the interpretation and validation of cluster analysis
- [3] D. L. Davies and D. W. Bouldin,
A Cluster Separation Measure,
Dostępne: <https://ieeexplore.ieee.org/document/9260048>