

NN - Raport

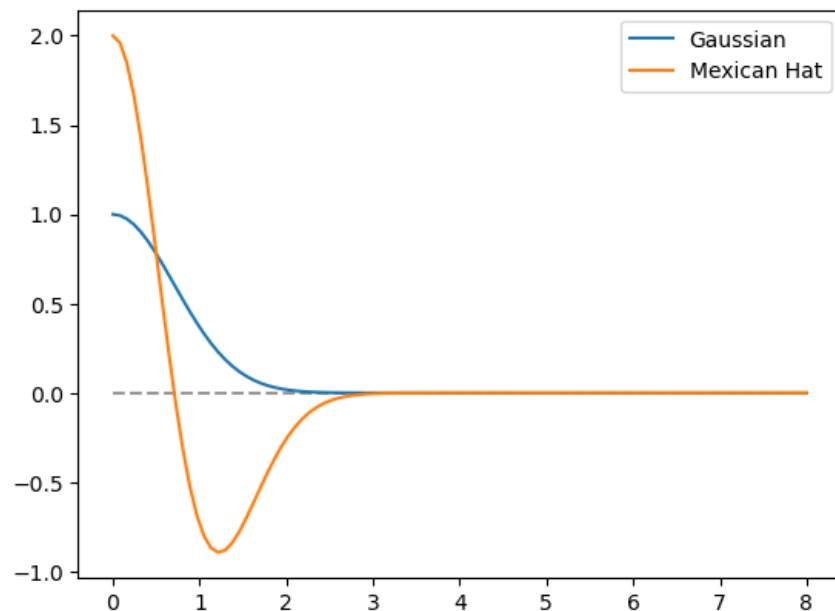
Aleksandra Samsel - 313396

1 Opis zadania

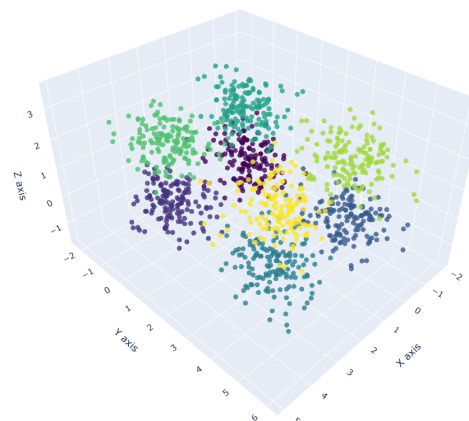
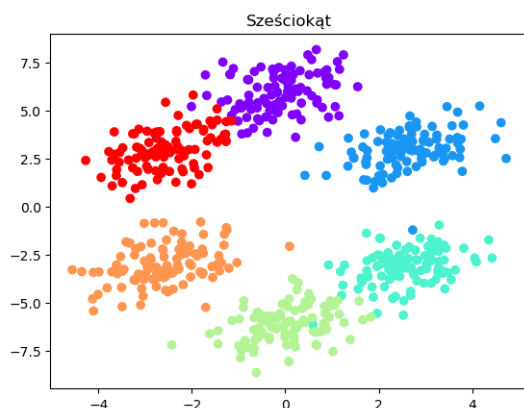
Raport dotyczy drugiego projektu, który polegał na implementacji sieci Kohonena. Sieć Kohonena (ang. SOM) jest to nienadzorowana technika uczenia maszynowego, która służy do tworzenia dwuwymiarowej reprezentacji wyższowymiarowego zbioru danych przy zachowaniu topologicznej struktury danych. Naszym zadaniem było dodać możliwość dostosowania parametrów: współczynnik uczenia, szerokość sąsiedztwa, rozmiar siatki oraz wybrać typ siatki i funkcję sąsiedztwa. Następnie na tak zaimplementowanej sieci przeprowadzone zostały eksperymenty porównujące różne architektury.

2 Podstawowa sieć Kohonena

Pierwszym zadaniem była implementacja sieci Kohonena na prostokątnej siatce. Użytkownik może wybrać parametry takie jak wymiary siatki, szerokość sąsiedztwa, współczynnik uczenia oraz funkcję sąsiedztwa. Przedstawione są one na poniższym wykresie.

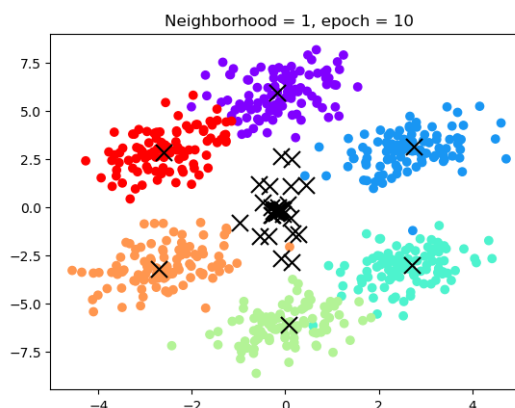
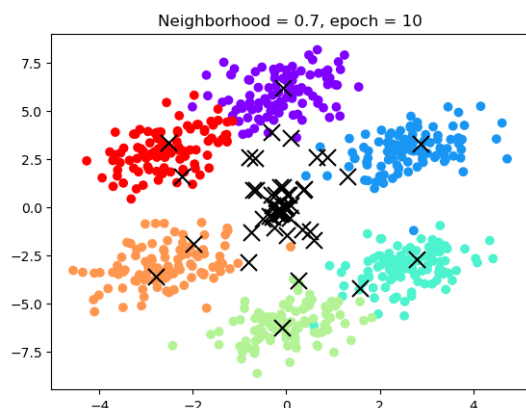
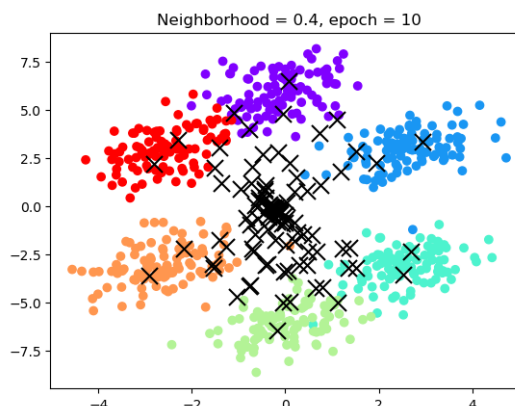
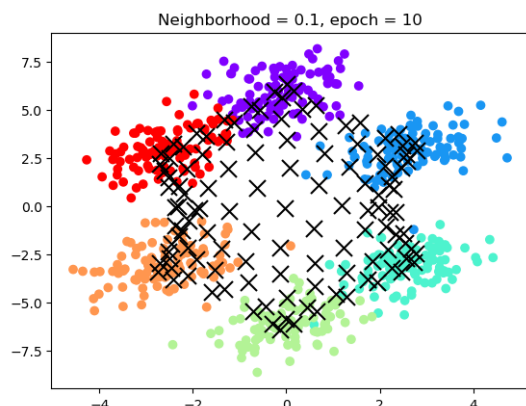


Do przetestowania działania naszej sieci zostały przygotowane 2 zestawy danych. Pierwszy z nich jest na płaszczyźnie 2 wymiarowej i przedstawia zbiory wierzchołków sześciokąta, drugi natomiast dotyczy punktów na płaszczyźnie 3 wymiarowej i są to zbiory wierzchołków sześciianu. Poniżej przedstawione zostały opisane zbiory:



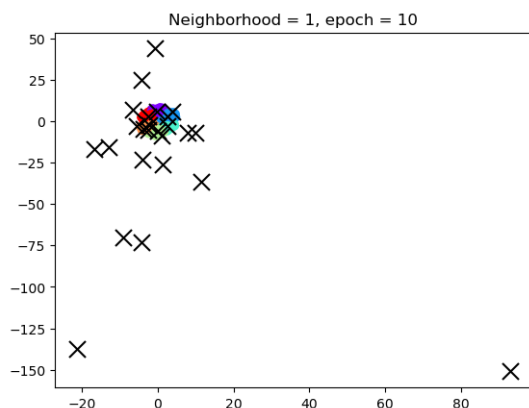
3 Sześciokąt

Pierwszymi testami było porównanie różnych wartości szerokości sąsiedztwa i jak wpływa to na klasteryzację. Wykonałam 4 testy dla następujących wartości: $[0.1, 0.4, 0.7, 1.0]$, wszystkie dla 10 epok, szybkości uczenia równej 0.01 i Gaussowskiej funkcji sąsiedztwa oraz siatki neuronów rozmiaru 11×11 . Poniżej przedstawione są wykresy. Najbardziej zadowalający dla nas wynik to szerokość sąsiedztwa równa 1.0, ponieważ 6 krzyżyków

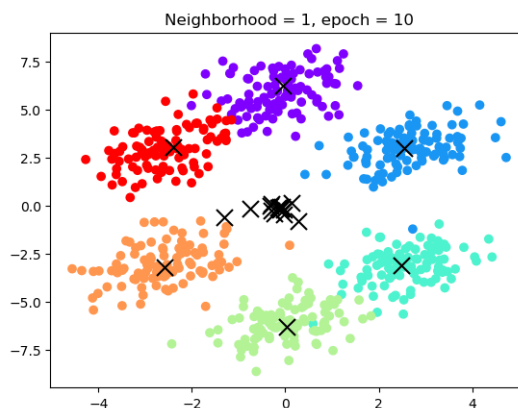


wyznacza nam 6 grup, które mamy a pozostałe grupują się w środku. Aby polepszyć ten wynik zastosujemy drugą funkcję sąsiedztwa - Mexican Hat. Stosując tę funkcję trzeba uważać na szybkość uczenia, bo możemy szybko otrzymać dużo krzyżyków wyrzuconych daleko poza interesujący nasz obszar z naszymi danymi, co możemy zobaczyć na poniższym wykresie (a). Po dobraniu odpowiednich parametrów otrzymujemy bardzo dobre dopasowanie, gdzie po zakryciu środkowej części wykresu otrzymujemy równo 6 grup pomimo siatki

neuronów wielkości 5×5 - wykres (b). Ostatnim testem jaki został wykonany w tej części raportu było

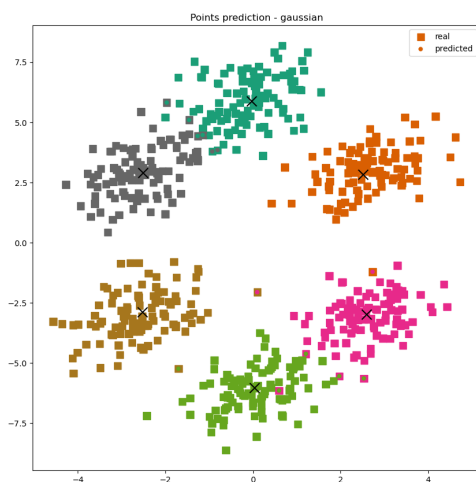


(a) Współczynnik uczenia = 0.02

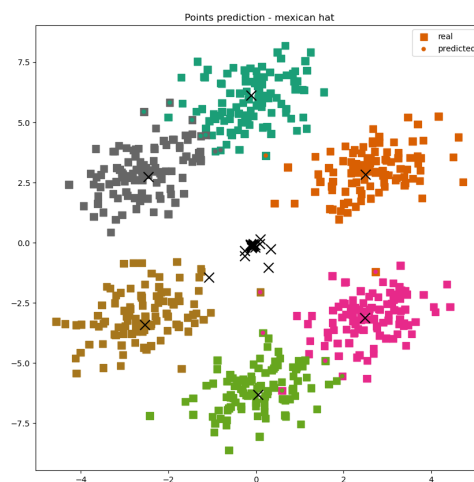


(b) Współczynnik uczenia = 0.003

sprawdzenie, czy punkty są klasyfikowane do dobrej grupy. Na poniższych wykresach kwadratami zostały przedstawione wartości prawdziwe, natomiast kółkami - predykcje. Każdy punkt, który ma różny kolor kwadratu i kółka jest źle zaklasyfikowany.



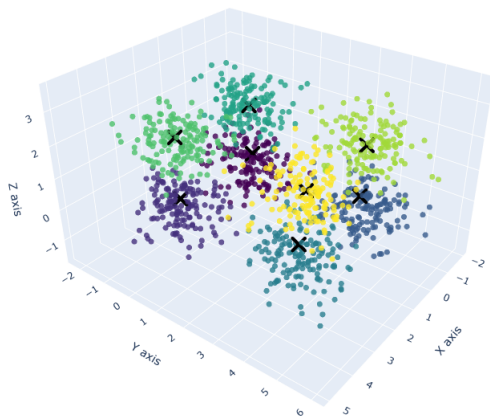
(a) Siatka 3×2



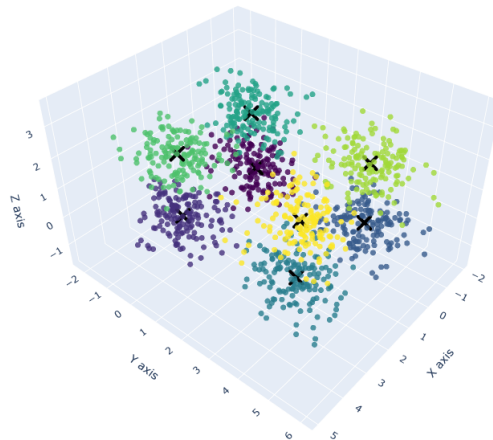
(b) Siatka 5×5

4 Sześcian

Dla danych zawierających wierzchołki sześcianu skupiłam się na otrzymaniu pojedynczego krzyżyka w każdej z grup oraz reszta krzyżyków miała skupiać się w jednym punkcie na środku. Gdy wiemy, że liczba klas to 8 - liczba wierzchołków sześcianu to w dość prosty sposób jesteśmy w stanie otrzymać zadowalające wyniki przedstawione poniżej dla obydwu funkcji sąsiedztwa.

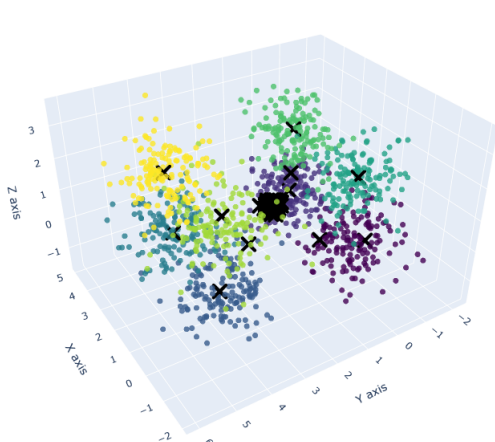


(a) funkcja Gaussowska

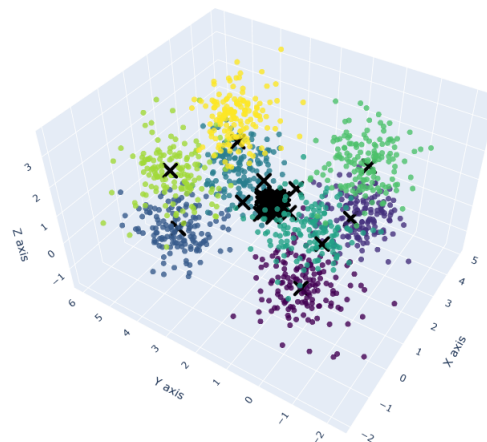


(b) funkcja Maxican Hat

Jednak w normalnych warunkach nie wiemy ile jest klas, dlatego dla N - liczba wierszy w danych $5\sqrt{N}$ wyznaczało liczbę nodów. Następnie starałam się stworzyć kwadratową siatkę, dlatego ostatecznie liczba nodów na jednym boku siatki wyznaczona była następująco: $\lfloor \sqrt{5\sqrt{N}} \rfloor$. Dla naszego zbioru danych oznaczało to siatkę o wymiarach 13×13 . W tym przypadku znacznie łatwiej osiągnąć porządną wynik dla funkcji Mexican Hat, co można zobaczyć na poniższych wykresach.



(a) funkcja Gaussowska



(b) funkcja Maxican Hat

5 Ulepszona(?) sieć Kohonena

Druga część projektu składała się z dodania sześciokątnej siatki oraz testów na nowych zbiorach danych. Pierwszym z ich jest zbiór danych *MNIST*, który zawiera zestaw cyfr w zapisie ręcznym. My używamy reprezentacji każdego piksela jako liczbę określającą odcień.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

Drugi zestaw danych to *Baza Danych Rozpoznawania Aktywności Ludzkiej* utworzona na podstawie nagrań 30 osób wykonujących codzienne czynności z zamontowanym na pasie smartfonem wyposażonym w wbudowane czujniki inercyjne.

5.1 MNIST

Doświadczenia zaczęłam od ustalenia najlepszych parametrów dla danych par funkcji sąsiedztwa i typu siatki. Przeprowadziłam 16 badań dla szerokości sąsiedztwa równych: 0.1, 0.4, 0.7, 1.0 oraz 16 dla współczynnika uczenia równego: 0.01, 0.001, 0.005, 0.0001. Wartości accuracy przedstawiłam poniżej:

Siatka	Funkcja	Współczynnik uczenia				Szerokość sąsiedztwa			
		0.01	0.001	0.005	0.0001	0.1	0.4	0.7	1.0
prostokątna	Gaussowska	0.4018	0.4078	0.3981	0.5308	0.4021	0.4634	0.467	0.399
	Mexican Hat	0.8865	0.8865	0.8865	0.4659	0.8865	0.8865	0.8865	0.7008
sześciokątna	Gaussowska	0.4116	0.5301	0.3943	0.5697	0.4842	0.4983	0.463	0.5031
	Mexican Hat	0.8865	0.8865	0.8865	0.5331	0.8865	0.8865	0.8837	0.792

Następnie dla wybranych najlepszych wyników (zaznaczonych na kolorowo w tabeli) zrobiłam po 3 testy dla sieci uczącej się 50 epok. Poniżej przedstawiam otrzymane średnie wyniki accuracy.

Siatka	Funkcja	Średnie accuracy
prostokątna	Gaussowska	0,5107
	Mexican Hat	0,8865
sześciokątna	Gaussowska	0,5971
	Mexican Hat	0,8865

Zdecydowanie funkcja sąsiedztwa Mexican Hat lepiej sobie poradziła z zadaniem. Jeśli chodzi o rodzaj siatki to nie ma dużej różnicy w wynikach, są trochę lepsze.

5.1.1 Human Activity Recognition

Dla drugiego zestawu danych zrobiłam analogiczne testy. Zaczęłam od wyboru najlepszej kombinacji współczynnika uczenia i szerokości sąsiedztwa. Zostały one wybrane na podstawie wyników accuracy, które zostały przedstawione w poniższej tabeli:

Siatka	Funkcja	Współczynnik uczenia				Szerokość sąsiedztwa			
		0.01	0.001	0.005	0.0001	0.1	0.4	0.7	1.0
prostokątna	Gaussowska	0.4178	0.3920	0.4163	0.6357	0.4249	0.3875	0.3937	0.4064
	Mexican Hat	0.8086	0.8086	0.8086	0.6226	0.8086	0.8086	0.8086	0.6433
sześciokątna	Gaussowska	0.3675	0.3924	0.3656	0.6371	0.4678	0.4779	0.3778	0.3713
	Mexican Hat	0.8086	0.8086	0.8086	0.6405	0.8086	0.8086	0.8086	0.8086

Następnie dla wybranych (zaznaczonych na kolorowo) najlepszych wartości przeprowadziłam potrójne testy dla poszczególnych architektur i przedstawiłam wyniki poniżej: Tak jak i testy dla poprzedniego zestawu

Siatka	Funkcja	Średnie accuracy
prostokątna	Gaussowska	0.4179
	Mexican Hat	0.8086
sześciokątna	Gaussowska	0.6341
	Mexican Hat	0.8086

danych, lepsza okazała się funkcja sąsiedztwa Mexican Hat a dodatkowo można zauważyć różnicę pomiędzy zastosowaniem siatki prostokątnej i siatki sześciokątnej - sześciokątna wypada lepiej.

6 Podsumowanie

Projekt pomógł stworzyć narzędzie służące do rozpoznawania klas oraz klasteryzacji. Udało nam się przetestować je na różnych zestawach danych i porównać wyniki. Ostatecznie można uznać, że funkcja Mexican Hat lepiej spełniła nasze oczekiwania - osiągała lepsze wyniki, jednak według mnie ciężiej było dopasować odpowiednie parametry aby osiągnąć zadowalające rezultaty.