Sieci Kohonena

Tomasz Makowski
nr indeksu 291132

Maj 2020

Spis treści

1 Wstęp

Celem tej pracy jest zbadanie wpływu różnych czynników na trenowanie sieci Kohonena.

W zaimplementowanej wersji sieci Kohonena można ustawiać dowolną wielkość siatki oraz zmieniać sposób ułożenia siatki (sześciokątne bądź kwadratowe sąsiedztwo). Do wyboru są również 2 funkcje liczenia odległości pomiędzy sąsiadami w siatce: odległość gaussa jak i minus druga pochodna odległości gaussa, czyli meksykański kapelusz.

2 Zbiory danych

W każdym eksperymencie wszystkie dane zaraz po wczytaniu były normalizowane. Każdy zbiór danych zawierał od 6 do 10 klas, które nie były używane podczas trenowania sieci. Jedynie po wytrenowaniu stanowiły one wielokolorowe tło do zaprezentowania wag sieci. Testy sieci zostały przeprowadzone na następujących 4 zbiorach danych:

1. hexagonal – zbiór zawiera 6 klas rozrzuconych w 6 okręgach, które są scentrowane w wierzchołkach sześciokąta (prawdopodobnie foremnego), jest to zbiór dwuwymiarowy zatem jest najłatwiejszy w wizualizacji oraz najlepiej widać na nim potencjalne dopasowanie sieci

2. cube – fikcyjny zbiór stworzony analogicznie do hexagonal, jednakże klasy są scentrowane w wierzchołkach sześciokąta, zatem jest ich 8, a zbiór ma 3 wymiary

3. mnist – jest to popularny zbiór danych zawierający zdjęcia ręcznie pisanych cyfr od 0 do 9. Zdjęcia są w rozmiarze 28x28 pikseli, zatem zbiór ma 784 wymiary. Każdy piksel zawiera wartość całkowitą od 0 do 256 oznaczającą natężenie koloru. Dane te zostały znormalizowane poprzez podzielenie wartości przez 256.

4. har – jest to zbiór różnych cech dotyczących zwykłych aktywności, np. chodzenia (w górę/ w dół), siedzenia, leżenia, stania. Te aktywności są klasami i jest ich 6. O każdej aktywności są różne informacje zbierane za pomocą czujników i są to wymiary danych. Jest ich 561. Zbiór danych był od razu znormalizowany na przedziale [-1, 1]

PCA

Do wizualizacji zbiorów danych 2-4 zostało użyte PCA. Dzięki temu możliwe było pokazanie wyników sieci wraz z prawdziwymi punktami na 2-wymiarowym wykresie.

3 Eksperymenty

3.1 Sposób liczenia odległości w siatce, parametr sąsiedztwa i liczba epok

3.1.1 Opis

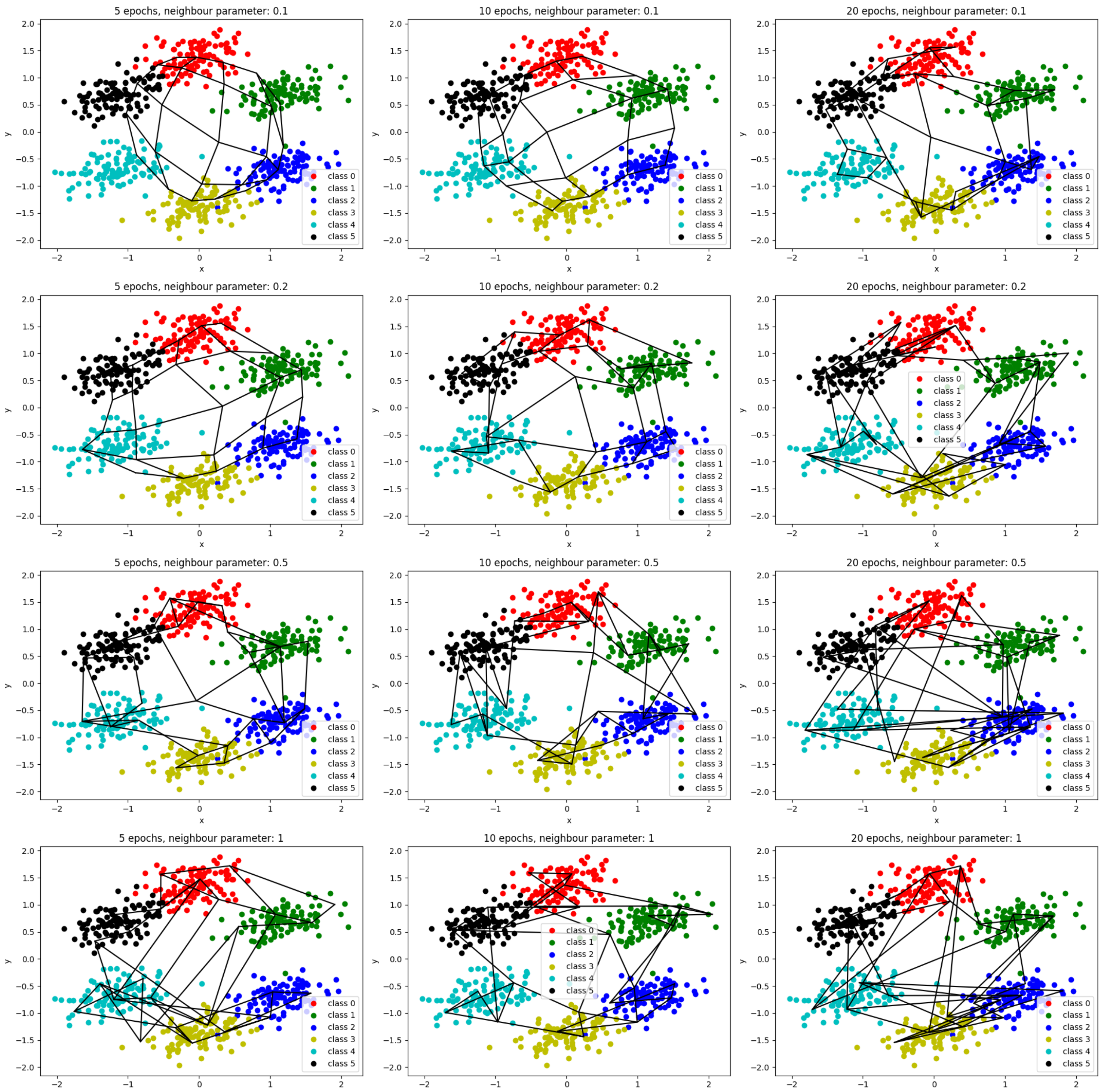
Pierwszym eksperymentem było sprawdzenie wpływu kilku czynników na jakość dopasowania sieci Kohonena do danych. Została sprawdzona liczba epok ze zbioru $\{5, 10, 20\}$, 2 funkcje do liczenia wpływu nowych danych na naukę sieci w zależności od odległości: Gauss i mexican hat (minus druga pochodna funkcji Gaussa), a także parametr sąsiedztwa. Parametr ten mnoży odległość pomiędzy danymi punktami na siatce przed zwróceniem wartości za pomocą funkcji gaussa bądź mexican hat. Parametr sąsiedztwa został zbadany spośród zbioru $\{0.1, 0.2, 0.5, 1\}$ dla funckji Gaussa oraz $\{0.005, 0.01, 0.015, 0.03\}$ dla funkcji mexican hat. Do sprawdzenia jakości wytrenowania sieci zostały użyte zbiory hexagonal i cube. W sieci użyto architektury opartej na kwadratowym sąsiedztwie oraz wymiarów siatki 5x5.

Na wykresach 1 i 2 zostało przedstawione porównanie dopasowania sieci w zależności od podanych parametrów dla funkcji gauss.

Natomiast na wykresach 3 i 4 zostało przedstawione porównanie dopasowania sieci w zależności od podanych parametrów dla funkcji mexican hat.

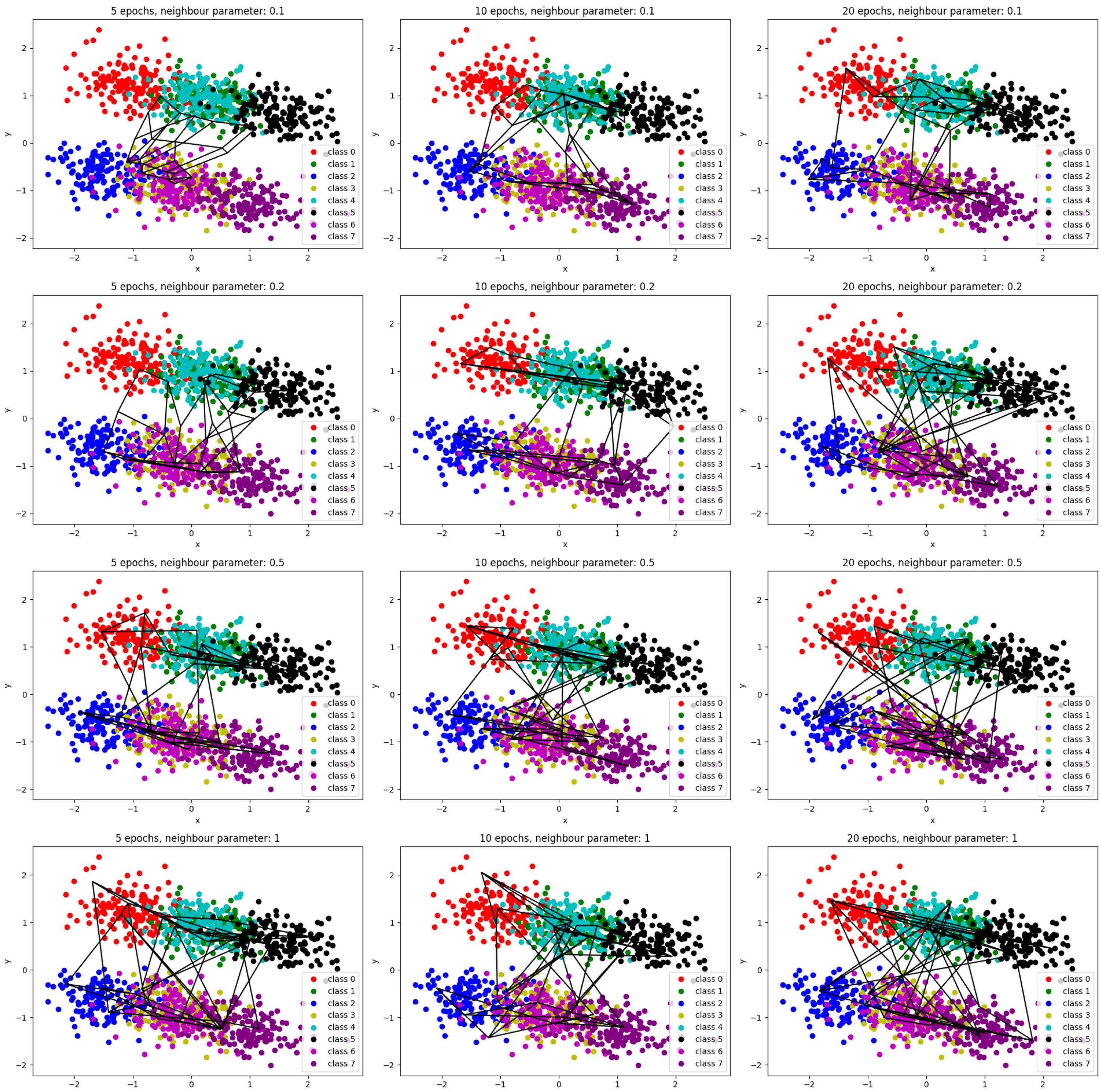
3.1.2 Wnioski

Na podstawie 1 można zaobserwować, że im mniej epok i im mniejszy parametr sąsiedztwa tym sieć jest bardziej niedouczona (np. 5 epok, parametr 0.1), natomiast dla odpowiednio dużych parametrów (np. 20 epok, parametr 1) sieć się przeucza i pojawiają się dziwne połączenia pomiędzy sąsiednimi neuronami – tak naprawdę sąsiednie neurony potrafią być oddalone od siebie. Reasumując dobrym optimum pomiędzy niedouczeniem się sieci, a przeuczeniem się sieci może być około 10 epok i parametr sąsiedztwa $0.2 - 0.5$. Wnioski te można również potwierdzić na wykresie 2 jednakże z uwagi na zastosowanie PCA tak naprawdę ciężko stwierdzić jakość dopasowania modelu do danych. W ogólności jedynie prawdziwe mogą być wnioski o przeuczeniu przy za dużych parametrach i niedouczeniu przy za małych parametrach.



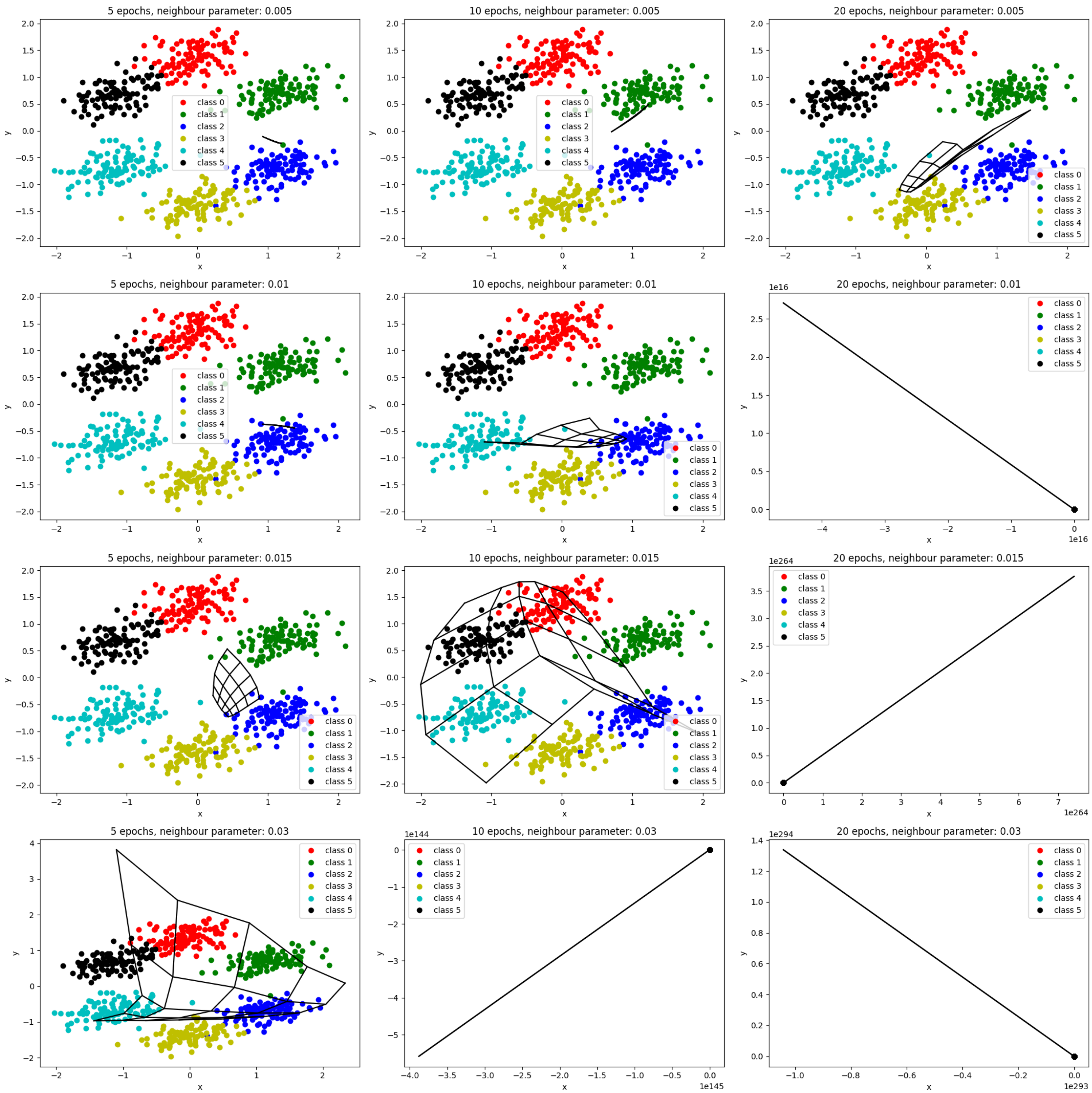
Rysunek 1: Hexagonal dataset, gauss function

W przypadku funkcji meksykański kapelusz ciężko mówić o jakimkolwiek nauczeniu się sieci. Za duże wartości parametrów powodują dążenie wartości neuronów do nieskończoności, natomiast za małe wartości powodują, że sieć się w ogóle nie uczy. Próbę nauczenia można jedynie zaobserwować w przypadku parametru



Rysunek 2: Cube dataset, gauss function

sąsiedztwa 0.015 i 10 epok oraz 0.03 i 5 epok. Jednakże w obu przypadkach można powiedzieć o tym, że sieć ma wagi rzędu wielkości danych co nie znaczy, że sieć w ogóle dopasowała się do klas.

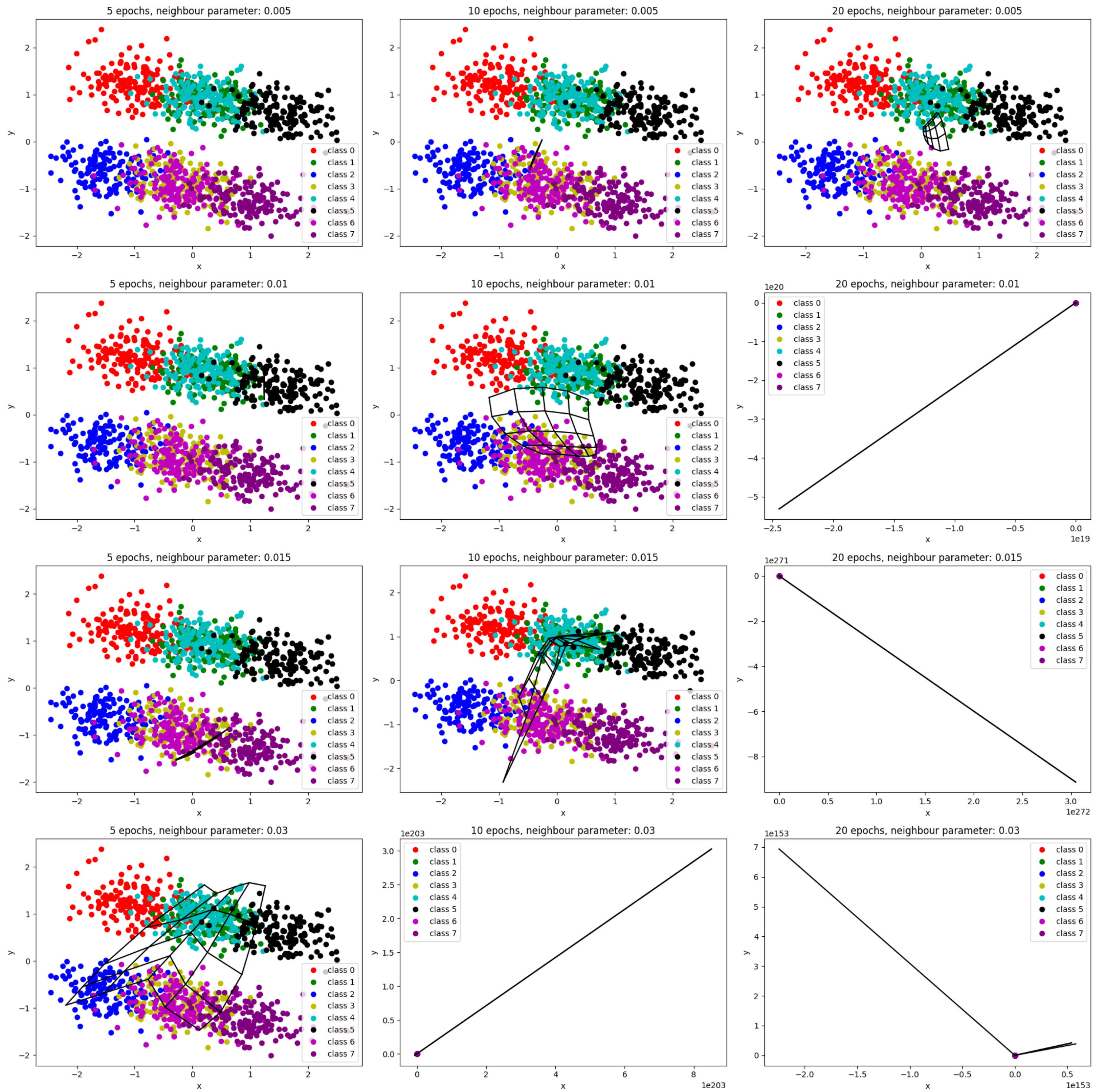


Rysunek 3: Hexagonal dataset, mexican hat function

3.2 Ułożenie neuronów w siatce

3.2.1 Opis

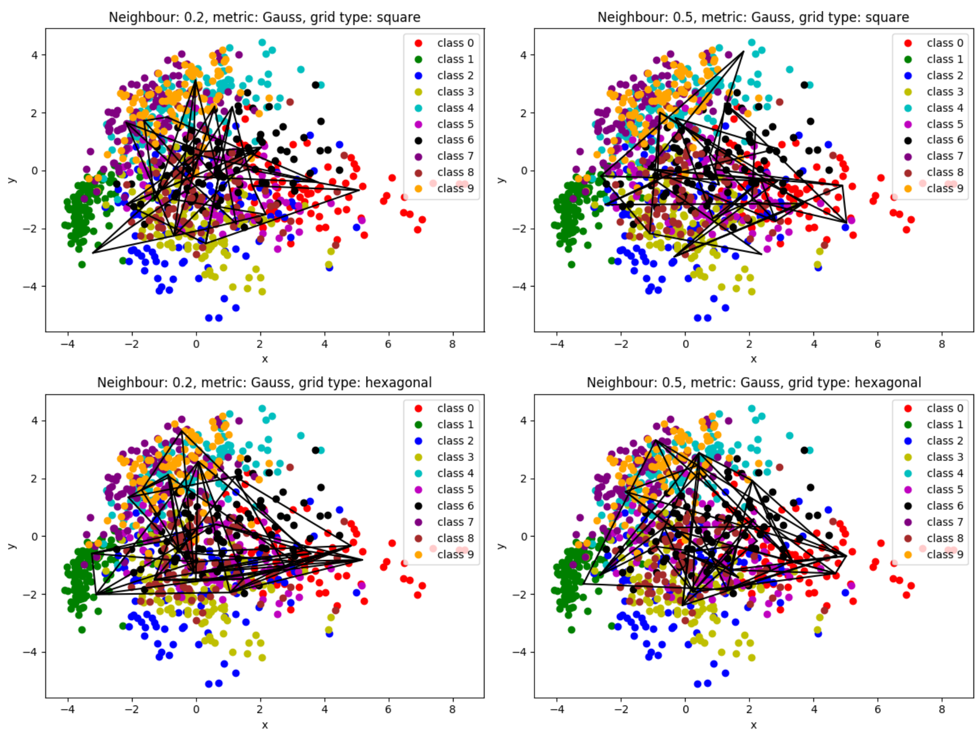
Drugim eksperymentem było sprawdzenie wpływu ułożenia neuronów w sieci. Dostępne były 2 architektury - ta co poprzednio, czyli ułożenie w siatce kwadratowej oraz w siatce sześciokątnej. W siatce szcześciokątnej



Rysunek 4: Cube dataset, mexican hat function

każdy neuron bezpośrednio sąsiaduje jedynie z 3 innymi neuronami co może dawać sieci pewną elastyczność. Oprócz ułożenia siatki zostały przetestowane 2 funkcje sąsiedztwa gauss i meksykański kapelusz. Dla gaussa zostały sprawdzone parametry sąsiedztwa ze zbioru: $\{0.2, 0.5\}$, natomiast dla meksykańskiego kapelusza zostały sprawdzone wartości: $\{0.01, 0.015\}$. Sieci były uczone na 10 epokach i miały wymiary 5x5. Do sprawdzenia parametrów użyto 2 zbiorów danych: har i mnist. Do wizualizacji zostało wykorzystane przekształcenie wielowymiarowych danych algorytmem PCA do 2 wymiarów.

Na wykresach 5 i 6 zostało przedstawione porównanie dopasowania sieci w zależności od podanych parametrów dla funkcji gauss.



Rysunek 5: Mnist dataset, gauss function

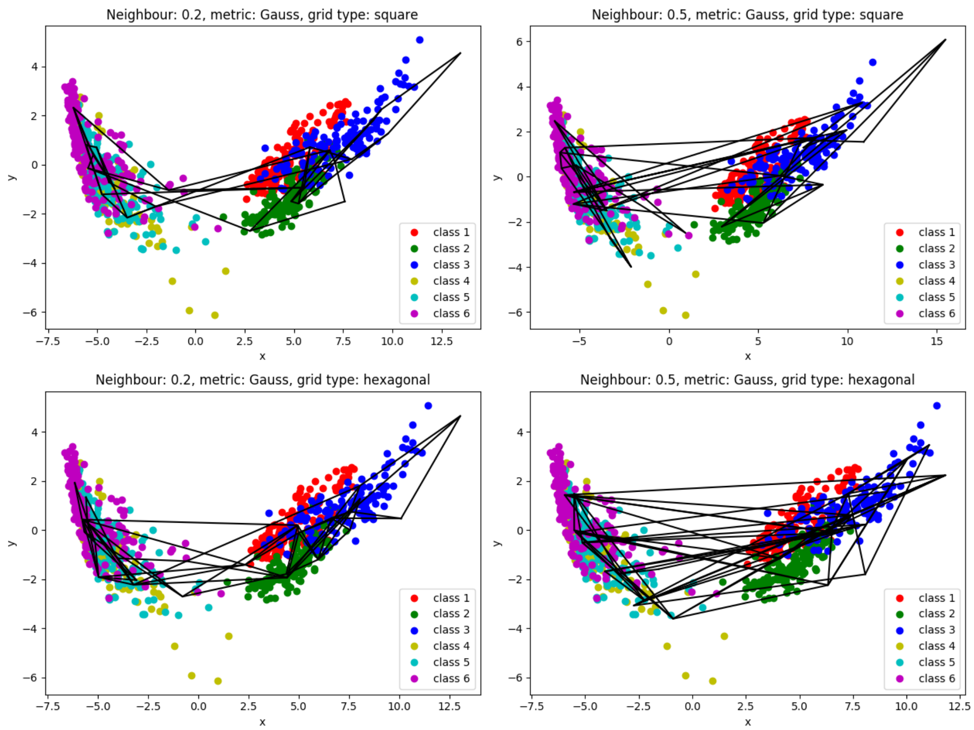
Natomiast na wykresach 7 i 8 zostało przedstawione porównanie dopasowania sieci w zależności od podanych parametrów dla funkcji mexican hat.

3.2.2 Wnioski

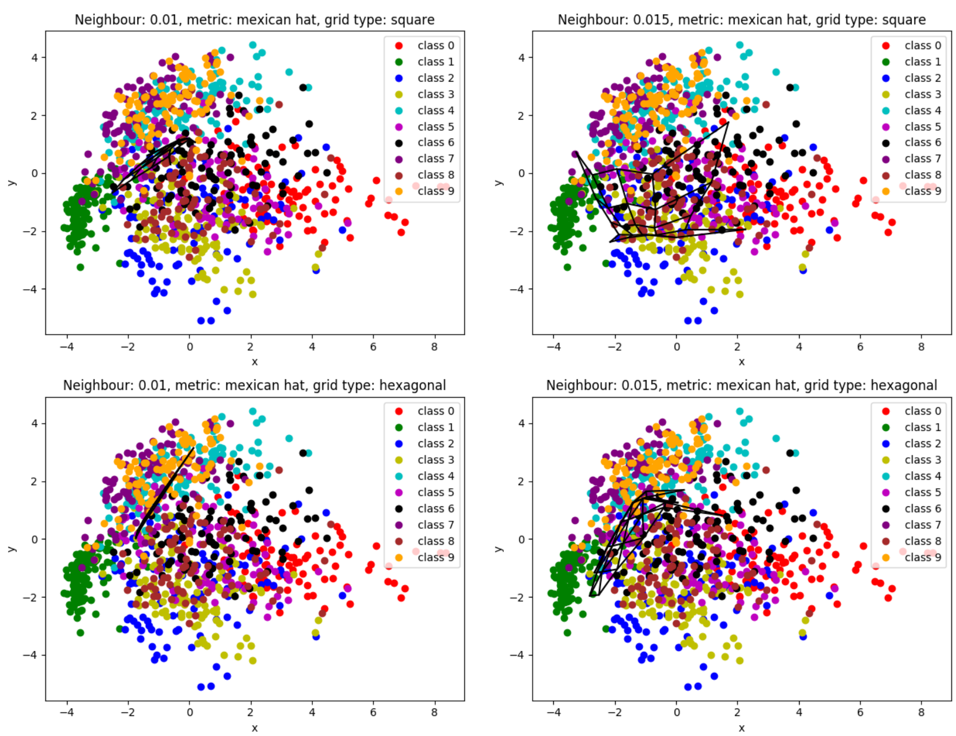
Na podstawie 5 i 6 można zaobserwować, że sieć w jakimś stopniu dopasowała się do danych. Niestety w przypadku MNIST nie da się jednoznacznie stwierdzić które podejście dało lepsze wyniki. Zmniejszanie wymiaru sieci za pomocą PCA powoduje nakładanie się neuronów i przeplatanie ze sobą siatki przez co ciężko w ogóle powiedzieć coś o jakości nauczonej sieci. W przypadku HAR można zaobserwować, że parametr sąsiedztwa 0.5 spowodował większy chaos w krawędziach siatki, zatem sieć prawdopodobnie się przeuczyła. Jeżeli chodzi o porównanie siatki sześciokątnej i kwadratowej nie da się jednoznacznie stwierdzić która siatka działa lepiej. Wpływ siatki jeżeli jest to jest niewielki.

Natomiast na wykresach 7 i 8 widać, że meksykański kapelusz nie pozwolił osiągnąć dopasowania do danych. Przy zbiorze MNIST sieć nie ma nawet wielkości zbioru danych. Dla zbioru HAR w większości przypadków zostały wykryte klasy 1-3 lub 4-6.

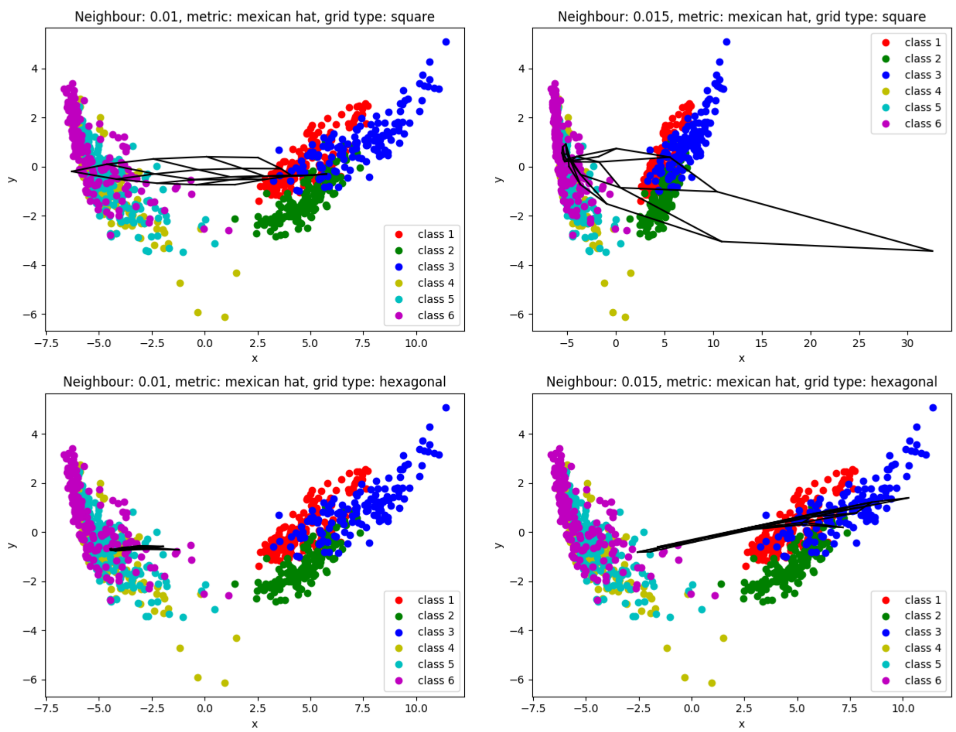
Zatem funkcja sąsiedztwa Gaussa jest lepsza niż meksykański kapelusz.



Rysunek 6: Har dataset, gauss function



Rysunek 7: Mnist dataset, gauss function



Rysunek 8: Har dataset, gauss function