

Raport: Sieci MLP

Tymoteusz Urban 320665

12 maja 2024

1 Wstęp

W niniejszym raporcie przedstawione zostały wyniki i analiza modelu sieci Kohonena. Model sieci został zaimplementowany od zera, bez użycia gotowych bibliotek implementujących rozwiązania typu DL i ML. Celem było zrozumienie działania sieci kohonena i wpływu parametrów na proces uczenia i podziału danych. We wszystkich eksperymentach była używana gaussowska funkcja sąsiedztwa, chyba że zostało stwierdzone inaczej.

2 KOH 1

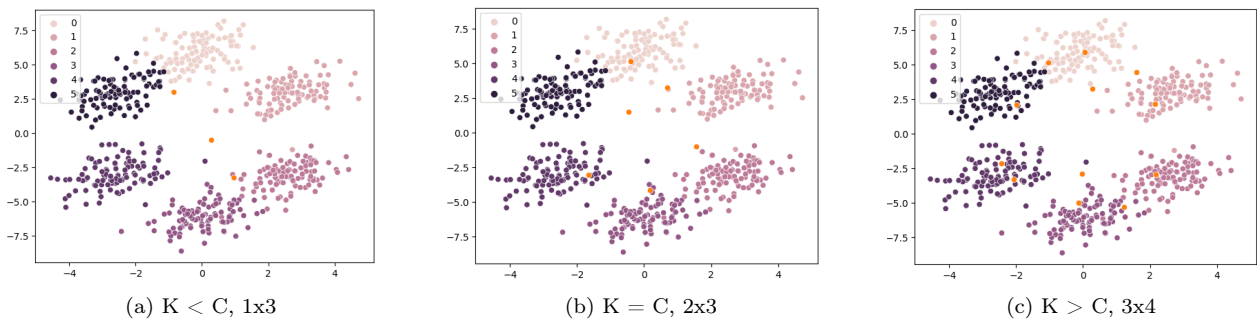
Pierwsze zadanie polegało na stworzeniu modelu sieci SOM, w której można ustawić rozmiar siatki, wybrać funkcję sąsiedztwa i ustawić parametry uczenia. Następnie należało przetestować sieć na dostarczonych zbiorach danych.

2.1 Zbiór danych Hexagon

Hexagon to zbiór danych 2d skupionych w wierzchołkach sześciokąta, zatem dobrym podziałem tych danych jest podział na 6 klastrów.

2.1.1 K vs C

Początkowo zostały sprawdzone różne warianty wartości $K = M * N$, gdzie $M \times N$ to rozmiar siatki, i wartości C oznaczającą prawdziwą liczbę klastrów.



Rysunek 1: Układ neuronów po 10 iteracjach

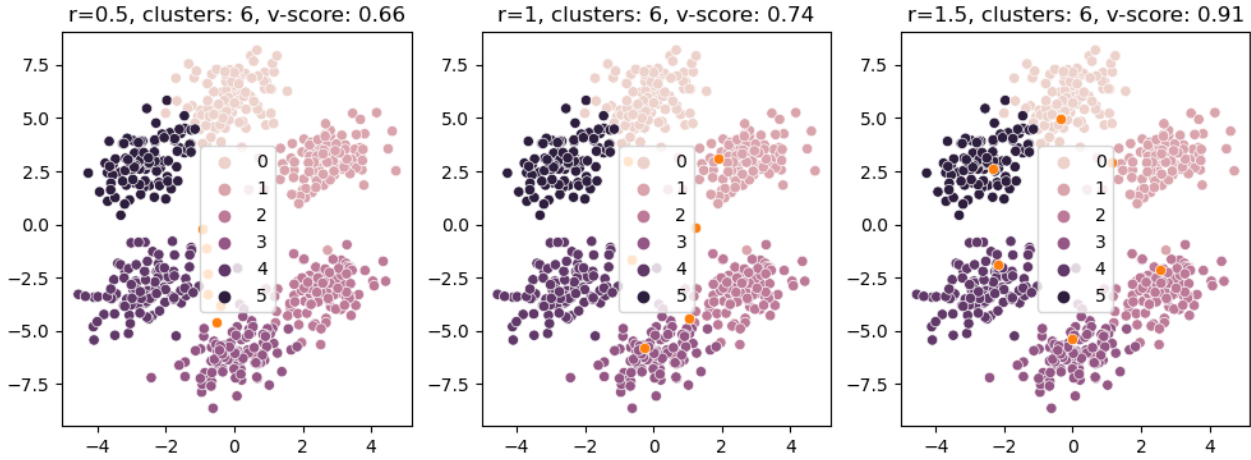
liczba neuronów	v measure
3	0.54
6	0.82
12	0.80

Tabela 1

Używanie liczby K mniejszej niż liczba klas jest bez sensu, gdyż nie mamy szans wykrycia wszystkich klastrów. V-measure wynosi jedynie 0.53. Użycie $K = C$ ma największy sens, gdyż wtedy każdy neuron odpowiada jednemu klastrowi. Widzimy że przy użyciu gęstszej siatki dostajemy dobre wyniki. Już przy 10 iteracjach punkty siatki osadzają się na klastrach. Normalnie nie znamy prawdziwej ilości klastrów, ale widzimy że lepiej zawyżać liczbę neuronów niż zaniżać.

2.1.2 Szerokość sąsiedztwa

Dodana została także możliwość zmiany szerokości sąsiedztwa za pomocą parametru r , przez który przemnażany jest argument x .

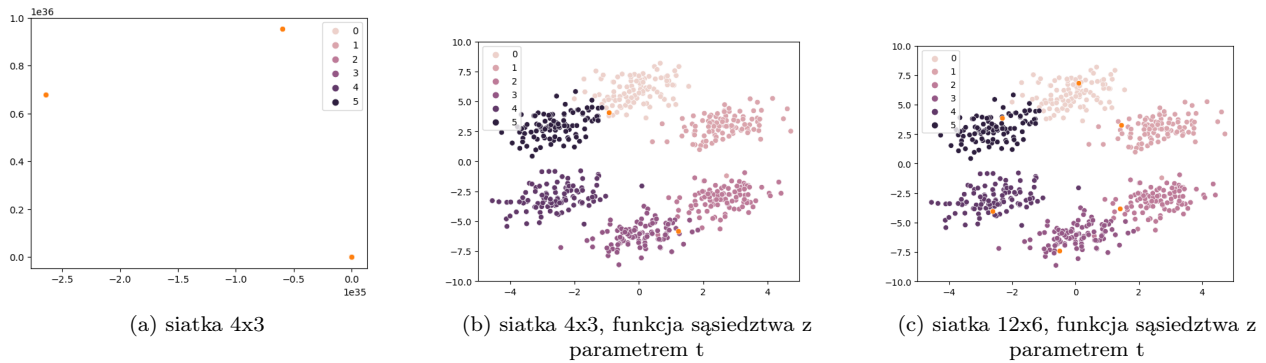


Rysunek 2: Eksperymenty

Widzimy, że zawężanie oddziaływanie funkcji przyciągania i ustawianie parametru r na większy od 1, daje lepsze wyniki klasteryzacji. Dla $r = 0.5$ punkty ustawiają się w jednym miejscu, podczas gdy dla $r = 1.5$ dobrze osadzają się na klastrach.

2.1.3 Meksykański kapelusz

Do danych zastosowana została także minus druga pochodną funkcji gaussowskiej jako funkcja sąsiedztwa.



Rysunek 3: Eksperymenty dla $\lambda_{max} = 20$

siatka	parametr t	liczba klastrów	v measure
4x3	nie	1	0
4x3	tak	2	0.52
12x6	tak	6	0.85

Tabela 2: Wyniki dla "meksykańskiego kapelusza"

Na domyślnych parametrach meksykański kapelusz jest bardzo niestabilny, wyrzuca punkty do ogromnych wartości. Korzystając jednak z neighborhood decay (parametr t w funkcji sąsiedztwa) funkcja zaczyna działać poprawnie. Nie dostajemy warningów, część neuronów została na miejscu, a v measure nie jest równy 0. Biorąc gęstszą siatkę otrzymujemy w zasadzie idealną klasteryzację.

W celu sprawdzenia czy nie był to przypadek, został przeprowadzony eksperyment. Rozmiar siatki był losowany w zakresie $[6-12] \times [6-12]$, zostało wytrenowanych 6 sieci:

	wynik
moda liczby klastrow	6
średni v measure	0.826(0.047)

Tabela 3: Wyniki dla "meksykańskiego kapelusza"

Widzimy zatem, że nie znając liczby klastrow możemy podać bardzo gęstą siatkę (ponad 10 razy więcej neuronów niż prawdziwych klastrow), a sieć i tak znajdzie tą optymalną liczbę wyrzucając resztę neuronów poza obręb danych.

2.2 Zbiór danych Cube

Zbiór danych Cube jest zbiorem punktów 3d skupionych w wierzchołkach sześcianu. W związku z tym prawdziwa liczba klastrow wynosi 8.

2.2.1 K vs C

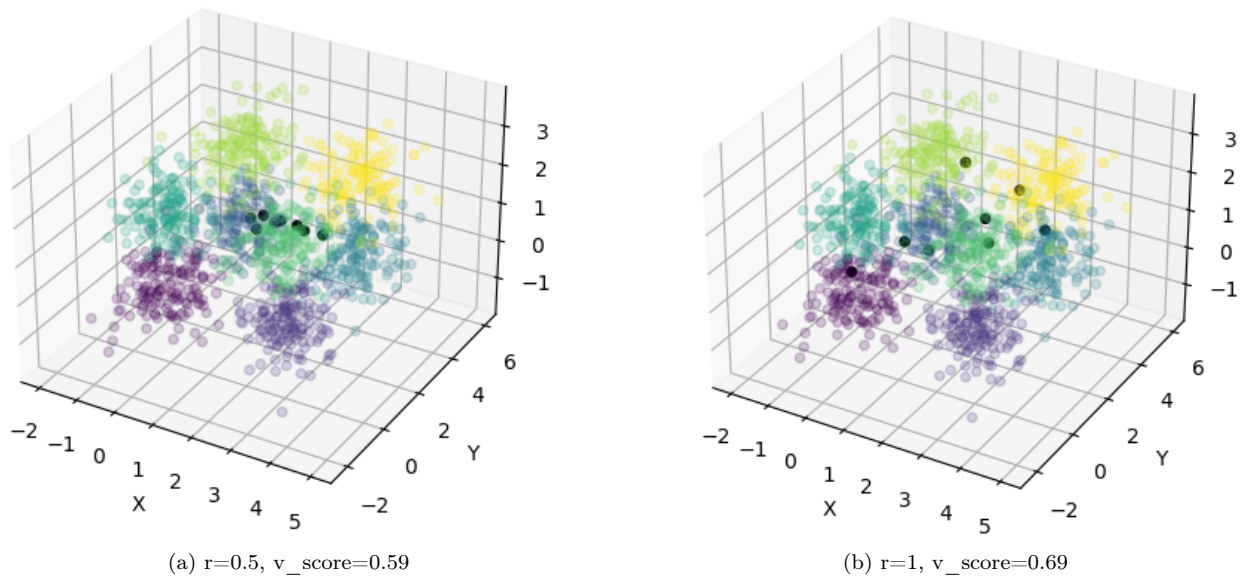
wariant	siatka	lambda	liczba klastrow	v measure
$K < C$	2x2	10	4	0.69
$K = C$	4x2	5	8	0.69
$K > C$	6x4	10	24	0.73

Tabela 4: Wyniki eksperymentów

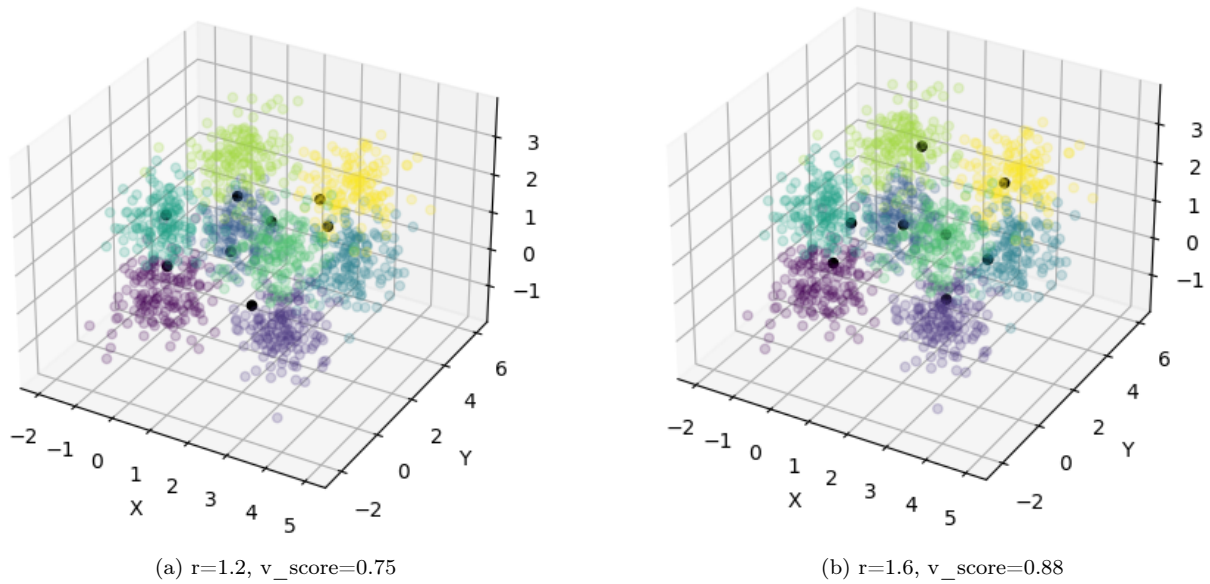
Jak widać wartości v measure nie są zbyt wysokie i dodatkowo są bardzo zbliżone dla wszystkich wariantów. Zobaczmy zatem czy manipulując szerokością sąsiedztwa możemy doprowadzić do lepszej klasteryzacji.

2.2.2 Szerokość sąsiedztwa

Wszystkie eksperymenty zostały przeprowadzone dla $\lambda_{max} = 5$ i rozmiaru siatki 4x2.



Rysunek 4: Eksperymenty część 1.



Rysunek 5: Eksperymenty część 2.

Łatwo wyciągnąć wniosek, że zawężanie oddziaływanie funkcji przyciągania i ustawianie parametru r na większy od 1, daje zdecydowanie lepsze wyniki klasteryzacji. Nie tylko v measure jest większy, ale na wykresie widać jak dobrze punkty układają się na klastrach.

2.2.3 Meksykański kapelusze

W celu przetestowania meksykańskiego kapelusza zostały wykonane testy jak ostatnio. Rozmiar siatki był losowany w zakresie $[6-12] \times [6-12]$, zostało wytrenowanych 6 sieci, używany był neighborhood decay, a λ_{max} była równa 20.

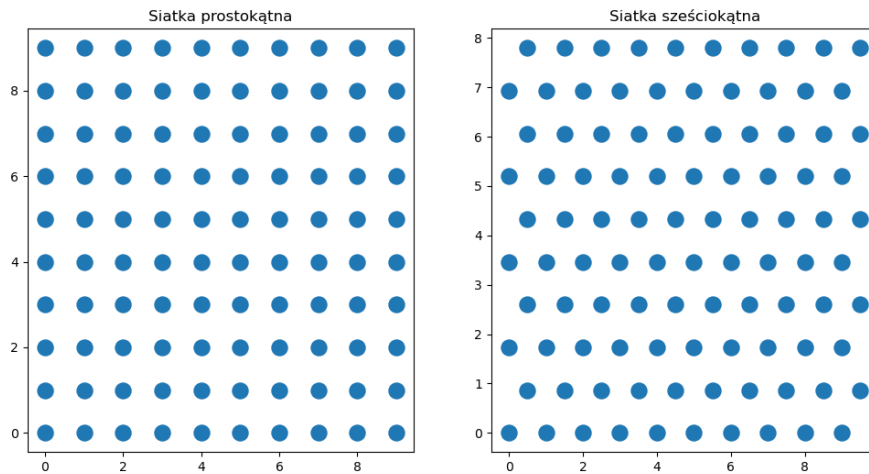
	wynik
moda liczby klastrów	6
średni v measure	0.747(0.061)

Tabela 5: Wyniki dla "meksykańskiego kapelusza"

Nienajgorzej, choć liczba klastrów jest nieco zaniżona.

3 KOH 2

Zadanie drugie polegało na dodaniu możliwości ułożenia neuronów w topologii siatki sześciokątnej. Dodatkowo należało przeprowadzić testy dla zbiorów MNIST i HARUS.



Rysunek 6: Topologie sieci

3.1 MNIST

Zbiór MNIST to zbiór danych, który składa się z tysięcy ręcznie zapisanych cyfr (od 0 do 9, w zatem liczba klastrów wynosi 10) o rozmiarze 28x28. W naszym przypadku dane są "spłaszczane" do jednowymiarowego wektora. Plan na eksperymenty jest następujący: wybrać lepszą topologię, następnie najlepszą funkcję sąsiedztwa i na znalezionej konfiguracji przeprowadzić eksperymenty z wartościami K i C.

topologia	średni v measure	odchylenie standardowe
prostokątna	0.34	0.03
sześciokątna	0.298	0.006

Tabela 6: Wyniki z 3 prób, siatka 2x5

Klasyczna siatka prostokątna daje lepsze wyniki. Jeśli chodzi o wybór funkcji sąsiedztwa to niestety, ale meksykański kapelusz się nie sprawdził. Pomimo wielu prób z różnymi wartościami parametrów (zmienianie szerokości sąsiedztwa, manipulowanie liczbą epok, korzystanie z decaya w funkcji sąsiedztwa, eksperymentowanie rozmiarem siatki, używanie bardzo małego learning rate) nie udało się przeprowadzić udanego treningu. Za każdym razem otrzymywana liczba klastrów wynosiła 1, a v measure 0.

Ostatecznie testy wartości K i C zostały przeprowadzone dla prostokątnej siatki i standardowej funkcji gaussowskiej.

wariant	siatka	v measure	odchylenie standardowe
$K < C$	3x2	0.22	0.03
$K = C$	2x5	0.34	0.03
$K > C$	8x8	0.45	0.04

Tabela 7: Wyniki z 3 prób

Otrzymane wyniki nie są najlepsze, ale ważne, że algorytm działa. Możemy zaobserwować to samo co w poprzednich eksperymentach - wraz ze wzrostem gęstości siatki polepszają nam się wyniki.

3.2 HARUS

Zbiór danych HARUS zawiera informacje z urządzeń typu wearables i służy do rozpoznawania i klasyfikowania aktywności wykonywanych przez ludzi, takich jak: chodzenie, wchodzenie po schodach, schodzenie po schodach, siedzenie, stanie, leżenie - razem 6 klastrów. Tutaj będziemy działać podobnie jak przy MNIST:

topologia	średni v measure	odchylenie standardowe
prostokątna	0.57	0.02
sześciokątna	0.623	0.009

Tabela 8: Wyniki z 3 prób, siatka 2x3, $\lambda_{max} = 10$

Tym razem topologia sześciokątna daje lepsze wyniki. Przejdźmy zatem do porównania funkcji sąsiedztwa.

funkcja	średni v measure	odchylenie standardowe	parametry
gauss	0.623	0.009	$\lambda_{max} = 10$
meksykański kapelusz	0.49	0.01	$\lambda_{max} = 5, lr=0.01, r=0.1$

Tabela 9: Wyniki z 3 prób

Dla meksykańskiego kapelusza zostały użyte dodatkowe parametry takie jak stała do zmiany szerokości sąsiedztwa (chcemy zatrzymać neurony w pobliżu danych) i nieduży learning rate (w celu zapobiegania uciekania wartości wag do nieskończoności). Ostatecznie testy wartości K i C zostały przeprowadzone dla sześciokątnej siatki i standardowej funkcji gaussowskiej.

wariant	siatka	v measure	odchylenie standardowe
$K < C$	1x3	0.53	0.01
$K = C$	2x3	0.61	0.02
$K > C$	6x4	0.52	0.07

Tabela 10: Wyniki z 5 prób

Dla zbioru HARUS klasteryzacja działa nieco lepiej niż dla MNIST. Ostatecznie najlepiej sprawdziła się klasteryzacja z liczbą neuronów równą liczbie klastrów.

3.3 Podsumowanie

Po wykonanych eksperymentach można dojść do wniosku, że w zadaniu wykrywania wzorców i klastrów w danych najlepiej korzystać ze standardowej funkcji gaussowskiej (meksykański kapelusz jest bardzo niestabilny), a jeśli nie znamy dokładnie liczby klastrów to lepiej stworzyć nieco gęstszą siatkę. Z kolei wybór topologii nie ma aż takiego wpływu na wyniki (mimo wszystko z założenia siatka sześciokątna powinna działać lepiej).