

# Metody inteligencji obliczeniowej - sieci Kohonena

Wiktor Wierzchowski

May 13, 2025

## 1 Wstęp

Poniższy raport jest podsumowaniem zadań wykonanych w ramach bloku tematycznego poświęconego sieciom kohonena, szerzej znanym w literaturze angielskojęzycznej jako mapy samoorganizujące się. Jest to rodzaj sieci neuronowej uczącej się w sposób nienadzorowany pozwalającej na przeprowadzanie klastrowania danych.

## 2 Zasada działania i implementacja

Struktura sieci kohonena oparta jest o zainicjowanie pewnej mapy neuronów. Neurony te są  $n$  wymiarowymi wektorami inicjowanymi losowo wokół centroidu danych gdzie ich wymiar odpowiada liczbie cech w danych. Oprócz ich konkretnej postaci funkcjonują one również w ramach pewnej topologii ustalonej przez postać siatki ich względnych położań. W implementacji wykonanej na potrzeby tego bloku przewidziane zostały dwa rodzaje siatek: prostokątna i heksagonalna.

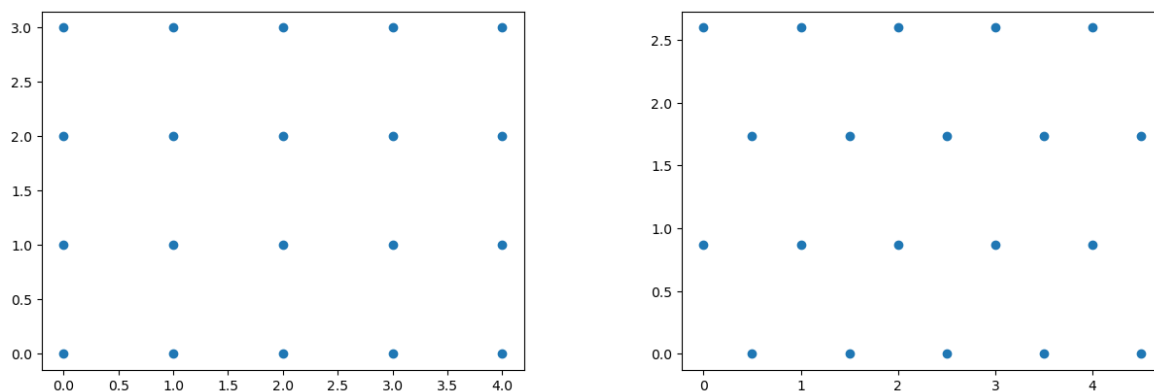


Figure 1: Przykład siatek w jakich inicjowana może być sieć Kohonena (lewa: prostokątna, prawa: heksagonalna).

Algorytm uczenia się sieci oparty jest na losowaniu obserwacji ze zbioru danych, odnajdywaniu neuronu najbardziej zbliżonego do rozważanej obecnie obserwacji określanego jako best matching unit (BMU), a następnie dalszym upodabnianiu BMU oraz otaczających go w siatce neuronów do tejże obserwacji. Ostatni krok uaktualniania wag jest miejscem, w którym postać siatki nabiera znaczenia. Sposób tej aktualizacji uwzględnia bowiem odległość każdego z neuronów do ustalonego BMU.

1. Losowe próbkowanie obserwacji -  $\hat{X} = rand(X)$
2. Ustalenie best matching unit -  $BMU = \arg \max_{i \in n, j \in m} ||\hat{X} - W_{ij}||$
3. Aktualizacja parametrów -  $\sigma(t) = \sigma_0 \exp(-\frac{t}{\lambda})$ , gdzie  $\lambda = \frac{n}{\log(\delta_0)}$  oraz  $\sigma_0 = \frac{\min(n, m)}{2}$

4. Ustalenie macierzy wpływu -  $H(i, j) = \text{gauss}(\|BMU - (i, j)\|, \sigma)$  lub  $H(i, j) = \text{ricker}(\|BMU - (i, j)\|, \sigma)$
5. Aktualizacja wag -  $W(t + 1) = W(t) + \alpha(t) \cdot H \cdot (\hat{X} - W(t))$

Gdzie  $m, n$  - wymiary siatki,  $W_{ij}$  - postać neuronu,  $\alpha(t)$  - krok,  $\text{gauss}(d, \sigma) = \exp(-\frac{d^2}{2\sigma^2})$ ,  $\text{ricker}(d, \sigma) = -\frac{d^2 - \sigma^2}{\sigma^4} \cdot \text{gauss}(d, \sigma)$ .

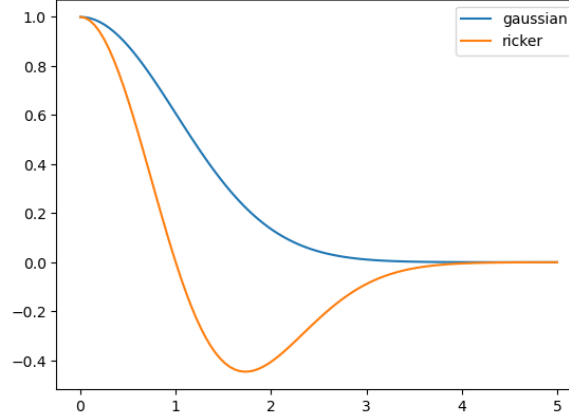


Figure 2: Funkcje wpływu dla zasięgu 1 ( $\sigma = 1$ ).

### 3 Testy na zbiorach danych

Poniżej przedstawione są testy sieci Kohonena jakie wykonane zostały na 4 różnych zbiorach. Pierwsze dwa zbiory są dwu i trzy wymiarowe co pozwala na graficzną reprezentację klastrowania i ostatecznych pozycji neuronów w sieci. Pozostałe zbiory mają więcej niż trójwymiarowe przestrzenie cech i wyniki zaprezentowane zostały przy użyciu macierzy błędów oraz oceny poprawności przez przyporządkowanie klastra do neuronów w oparciu o otaczające go obserwacje.

#### 3.1 Zbiór danych heksagon

Zbiór danych o dwóch cechach. Liczba obserwacji 600.

- Siatka: heksagonalna, 2x3
- Uczenie: iteracje = 5, krok początkowy = 0.2, zasięg wpływu początkowy = 2

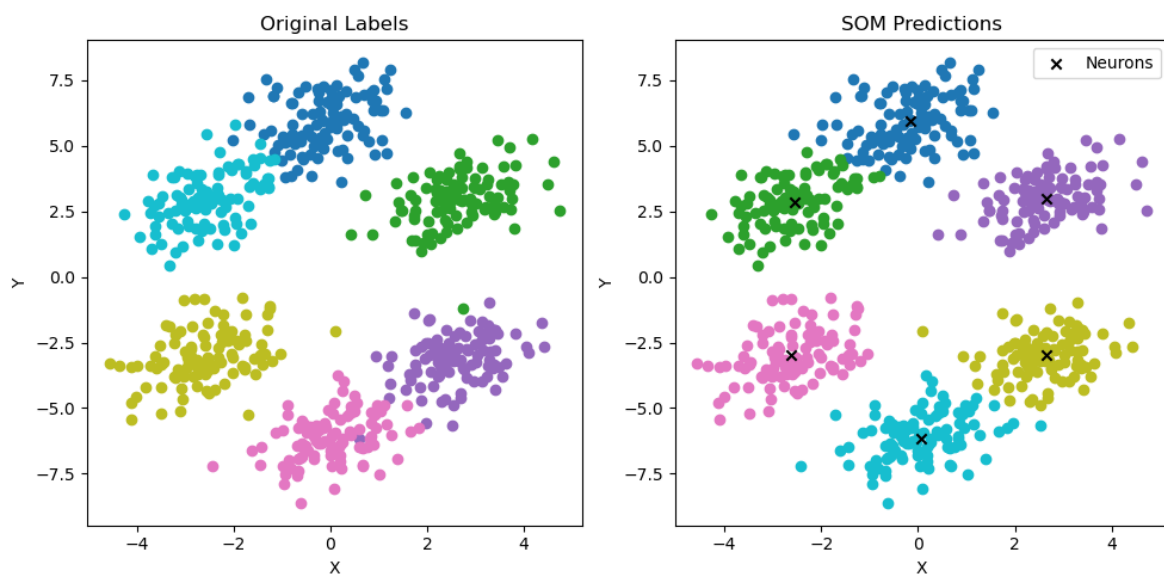


Figure 3: Zbiór heksagon (lewa). Wynik klastrowania zbioru heksagon siecią kohonena (prawa).

### 3.2 Zbiór danych cube

Zbiór danych o trzech cechach. Liczba obserwacji 1200.

- Siatka: prostokątna, 4x2
- Uczenie: iteracje = 10, krok początkowy = 0.1, zasięg wpływu początkowy = 2

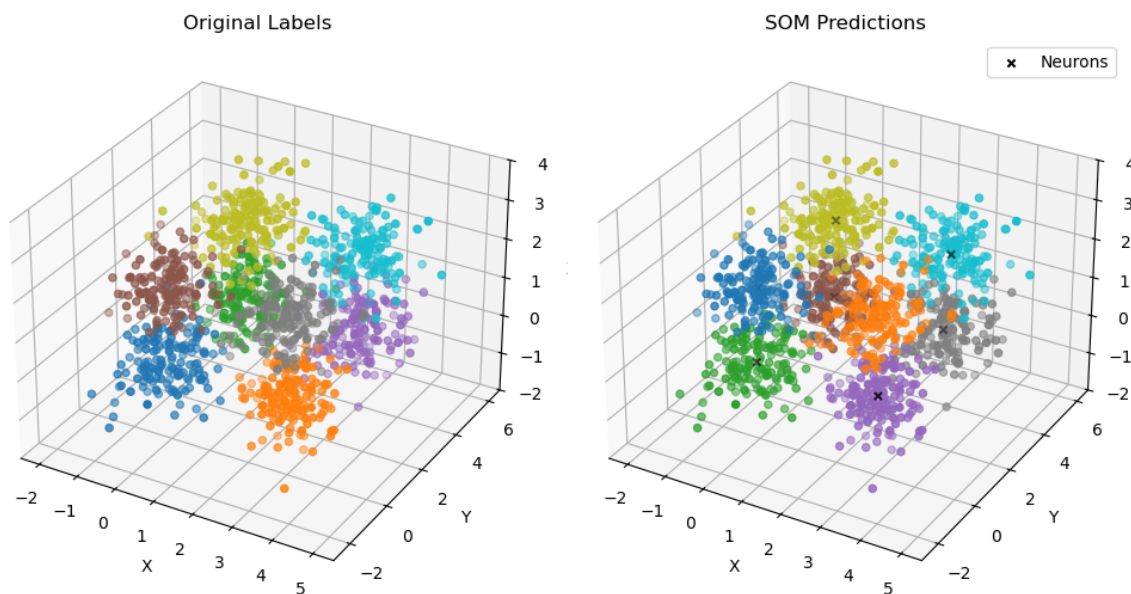


Figure 4: Zbiór cube (lewa). Wynik klastrowania zbioru heksagon siecią kohonena (prawa).

### 3.3 Zbiór danych MNIST

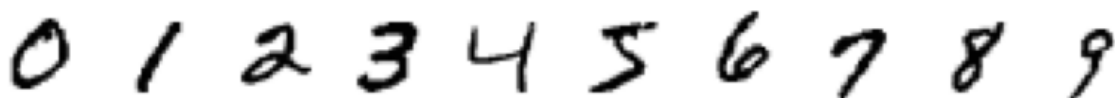


Figure 5: Przykładowe obserwacje ze zbioru MNIST.

Zbiór danych o 784 cechach. Liczba obserwacji 70000.

- Siatka: prostokątna, 9x1
- Uczenie: iteracje = 50, krok początkowy = 0.1, zasięg wpływu początkowy = 4

	(np.int64(0), np.int64(0))	(np.int64(0), np.int64(1))	(np.int64(0), np.int64(2))	(np.int64(0), np.int64(3))	(np.int64(0), np.int64(4))	(np.int64(0), np.int64(5))	(np.int64(0), np.int64(6))	(np.int64(0), np.int64(7))	(np.int64(0), np.int64(8))
0	7	22	748	19	42	247	44	5478	296
1	4300	3528	8	10	6	6	9	0	10
2	449	425	154	87	206	217	4933	69	450
3	532	69	1198	60	189	64	259	35	4735
4	213	292	46	2309	3698	205	46	14	1
5	262	635	2322	289	441	126	17	68	2153
6	395	145	288	3	97	5693	126	92	37
7	385	335	11	4456	2018	5	57	22	4
8	399	348	3593	282	257	57	61	38	1790
9	288	109	81	2907	3384	14	20	50	105

Figure 6: Wynik klastrowania zbioru MNIST siecią kohonena.

### 3.4 Zbiór danych HARUS

Zbiór danych o 561 cechach. Liczba obserwacji 7352.

- Siatka: heksagonalna, 3x2
- Uczenie: iteracje = 25, krok początkowy = 0.1, zasięg wpływu początkowy = 4

	(np.int64(0), np.int64(0))	(np.int64(0), np.int64(1))	(np.int64(0), np.int64(2))	(np.int64(1), np.int64(0))	(np.int64(1), np.int64(1))	(np.int64(1), np.int64(2))	(np.int64(2), np.int64(0))	(np.int64(2), np.int64(1))	(np.int64(2), np.int64(2))
-1	24	27	9	6	45	18	4	29	27
0	1258	1243	250	318	1974	617	247	712	540
1	0	1	0	0	2	0	0	1	0

Figure 7: Wynik klastrowania zbioru HARUS siecią kohonena.

## 4 Wnioski

Sieci kohonena są mocno wrażliwe na wielkość siatki, a więc liczbę neuronów biorących udział w procesie uczenia. W sytuacji gdy nie znamy liczby klastrów sieć skazana jest na wskazanie większej ich liczby niż jest obecna. Rozwiązaniem tego może być przypisywanie klastrów do neuronów w sposób uwarunkowany otoczeniem neuronu. Gdy proces uczenia jest już zakończony możemy ustalić, w oparciu o etykiety większości obserwacji w otoczeniu każdego z neuronów, do jakiego klastra dany neuron

przynależy. Niestety w świecie rzeczywistym gdzie nie mamy możliwości podejrzenia do jakiej etykiety powinny zostać przypisane nasze obserwacje, takiej możliwości nie ma. Sieci Kohonena są bardzo ciekawym conceptualnie narzędziem ale w praktyce korzystanie z nim sprowadza się do cierpliwego manipulowania parametrami w nadziei, że któraś z odpowiedzi jakie uzyskamy będzie bliska prawdzie.