

SOM Self Orgnizing Map

Samoorganizacja i uczenie konkurencyjne

Uczenie nadzorowane i nienadzorowane

Uczenie nadzorowane:

- sygnał wejściowy $\mathbf{x}_i$ posiada oczekiwaną odpowiedź $\mathbf{y}$
- trening dąży do minimalizacji błędu pomiędzy odpowiedzią sieci a wartością oczekiwaną
- klasyfikacja, aproksymacja, ...

Uczenie nienadzorowane:

- modelowanie danych $\mathbf{x}_i$ (brak sygnału zwrotnego $\mathbf{y}_i$)
- samoorganizacja - prowadzący do uporządkowania sieci poprzez lokalne, niezależne oddziaływania

Mechanizmy samoorganizacji

- oparte o regułę asocjacji Hebba, metody korelacyjne
- oparte o mechanizm konkurencji - reguła Kohonena

Aspekty samoorganizacji

- trening dostosowuje wagi neuronów (a przez to ich aktywacje) do aktywności sygnałów uczących
- pewne grupy neuronów reagują na podobne wzorce
- większe sygnały -> większe wagi -> większe aktywacje -> większe różnice pomiędzy grupami neuronów reagujących na odmienne bodźce
- nadmiarowość danych (redundancja) - wielokrotne powtarzanie podobnych wzorców jest niezbędne w samoorganizacji

Samoorganizacja i uczenie nienadzorowane

Samoorganizujące się sieci neuronowe

- wykrywanie istotnych cech w sygnale, podobieństw i różnic
- modelowanie rozkładu danych
- analiza skupień (klasteryzacja) - grupowanie wektorów podobnych
- analiza składowych głównych, redukcja wymiarowości do najistotniejszych cech
- prototypy - wykrycie typowego reprezentanta pewnej grupy wektorów
- wektorowa kwantyzacja, kodowanie, kompresja - opisanie zbioru danych mniejszą liczbą wektorów kodujących (prototypów)
- mapy cech - organizacja danych w niskowymiarowych przestrzeniach (1D, 3D, 3D) przy zachowaniu porządku topologicznego

Uczenie konkurencyjne

- **Uczenie konkurencyjne** (*competitive learning*) - w trakcie uczenia neurony konkurują ze sobą o prawo do reprezentacji danych wejściowych
- Zwycięzca bierze wszystko WTA (*winner takes all*) nazywane również Hard Competitive Learning, tylko sygnał neuronu najbardziej pobudzonego (zwycięzcy) jest brany pod uwagę
- Zwycięzca bierze większość WTM (*winner take most*) - łagodniejsza forma biorąca pod uwagę neurony o słabszym pobudzeniu, w pewnym otoczeniu $h(\mathbf{r}_i, \mathbf{r})$
- Zwykle zakładamy unormowanie wektorów wag $\mathbf{w}$ lub sygnału wejściowego $\mathbf{x}$

Neuron zwycięzca k o największym pobudzeniu

$$\mathbf{w}_k^T \mathbf{x} > \mathbf{w}_j^T \mathbf{x}, \quad j = 1, \dots, N$$

Dla znormalizowanych wektorów $\mathbf{w}_i$ oraz $\mathbf{x}$

$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

neuron zwycięzca $\mathbf{w}_k$ ma najmniejszą odległość do $\mathbf{x}$

$$\|\mathbf{x} - \mathbf{w}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{w}\|^2 - 2\mathbf{w}^T \mathbf{x}$$

$$\min \|\mathbf{x} - \mathbf{w}\| = \max \mathbf{w}^T \mathbf{x}$$

stąd neuron zwycięzca

$$k = \arg \min_j \|\mathbf{w}_j - \mathbf{x}\| = \sqrt{\sum_{i=1}^d (x_i - w_{ij})^2}, \quad j = 1, \dots, N$$

Miary podobieństwa

- odległość Euklidesowa L_2

$$d(\mathbf{x}, \mathbf{w}_j) = \sqrt{\sum_{k=1}^d (x_k - w_{jk})^2}$$

- odległość Manhattan L_1

$$d(\mathbf{x}, \mathbf{w}_j) = \sqrt{\sum_{k=1}^N |x_k - w_{jk}|}$$

- iloczyn skalarny (odległość kątowa)

$$d(\mathbf{x}, \mathbf{w}_j) = 1 - \mathbf{w}_j^T \mathbf{x} = 1 - \|\mathbf{x}\| \|\mathbf{w}_j\| \cos(\mathbf{x}, \mathbf{w}_j)$$

- norma L_∞

$$d(x, w_j) = \max_k |x_k - w_{jk}|$$

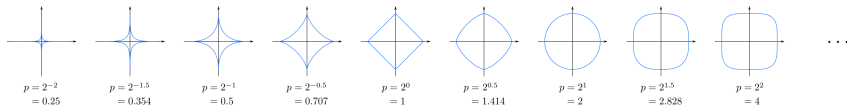
Odległość Minkowskiego

Uogólniona funkcja odległości

$$d(x, y) = \left(\sum_{k=1}^K (x_k - y_k)^p \right)^{\frac{1}{p}}$$

- $p = 1$ odległość Manhattan
- $p = 2$ odległość euklidesowa

Jednostkowy okrąg w przestrzeni R^2



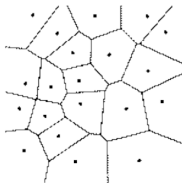
Rys: [wikipedia.org](https://en.wikipedia.org/wiki/Minkowski_norm)

Voronoi i Delaunay

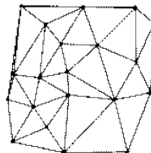
Punkty danych



granice decyzji
Voronoi



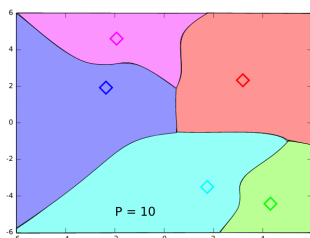
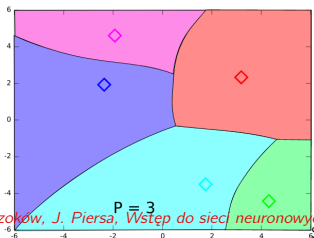
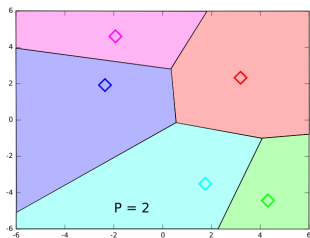
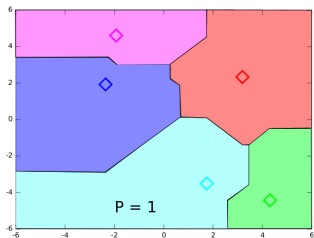
Triangulacja
Delauny



Neurony dzielą przestrzeń cech na rozdzielne obszary (obszary Voronoi), w danym obszarze tylko jeden neuron zwycięża konkurencję

- Zbiór Voronoia - zbiór wektorów wewnątrz obszaru Voronoia.
- Łącząc neurony, których obszary Voronoia mają wspólną krawędź otrzymujemy triangulację Delaunaya.

Strefy wpływów

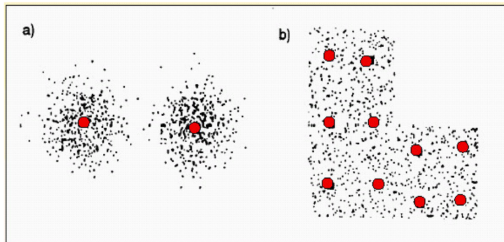


Rys: M. Czokow, J. Piersa, Wstęp do sieci neuronowych

Sieci Neronowe

Kwantowanie wektorowe

- **Kwantowanie wektorowe** - zakodowanie danych za pomocą wektorów prototypowych. Pojedynczy prototyp reprezentuje grupę wektorów wejściowych
- Księga kodów - zbiór prototypów kodujących (ich numery)
- Kompresja danych - liczba prototypów jest mniejsza od liczby wzorców



Algorytm kwantowania wektorowego WTA

1. Zainicjuj wagi k neuronów
2. Dla każdego wzorca uczącego $\mathbf{x}$ znajdź neuron zwycięzcę o największej aktywacji
3. Przesuń wagi zwycięskiego neuronu w kierunku wektora $\mathbf{x}$ (reguła Kohonena)

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta(t)\|\mathbf{x} - \mathbf{w}(t)\|$$

4. Powtarzaj 2 i 3 aż do uzyskania zbieżności

Błąd kwantowania wektorowego dla k neuronów

$$E(\mathbf{x}; \mathbf{w}, t) = \frac{1}{k} \sum_i \|\mathbf{x}_i - \mathbf{w}_{m(i)}\|^2$$

gdzie $\mathbf{w}_{m(i)}$ to wagi neuronu zwycięskiego dla sygnału $\mathbf{x}_i$

Normalizacja wektorów

- Proces samoorganizacji prowadzi do spójnego podziału przestrzeni cech, gdy wektor $\mathbf{w}$ jest znormalizowany (po każdej adaptacji poddajemy go normalizacji)
- Gdy wektory $\mathbf{x}$ są znormalizowane, wówczas $\mathbf{w}_j$ również w czasie treningu będą stawały się automatycznie znormalizowane
- normalizacja

$$\hat{x}_i = \frac{x_i}{\sqrt{\sum_{i=1}^d x_i^2}}$$

- lub poprzez dodanie dodatkowego wejścia x_0 spełniającego

$$\sum_{i=0}^d x_i = 1$$

Przykład: kompresja obrazów

- obraz o wymiarach $N_x \times N_y$ dzielimy na równe ramki o wymiarach $n_x \times n_y$
- wektor $\mathbf{x}_i$ to wektor $n_x \times n_y$ pikseli (np. odcienie szarości)
- uczenie samoorganizujące dowolna metoda w celu minimalizacji błędu kwantyzacji
- podobne ramki pobudzają ten sam neuron zwycięski
- prezentacja wszystkich ramek $\mathbf{x}_i$ pozwala ustalić kolejność neuronów zwycięskich 1, 5, 1, 30, ..., ich numery tworzą księgę kodów
- współczynnik kompresji

$$k_r = \frac{N_r n_x n_y T}{N_r \log_2 n + n n_x n_y t}$$

gdzie N_r liczba ramek, n - liczba neuronów, T liczba bitów potrzebna do reprezentacji wartości x_i , t liczba bitów potrzebna do reprezentacji wagi w_i

original



VQ 1024 codebook



👉 Image Compression with Vector Quantization by Ivan-Assen Ivanov

Mapy ekwiprobabilistyczne

- Zasada jednakowego zniekształcenia: każdy neuron powinien mieć podobny wkład do końcowego błędu kwantyzacji (klasyfikacji).
- przecenia rejony mało prawdopodobnych danych i niedojenja rejony o dużym prawdopodobieństwie
- przy losowej minimalizacji część neuronów może znaleźć się w obszarach gdzie nie będzie pobudzeń (martwe neurony)
- powstają neurony bezużyteczne; jak wykorzystać całą sieć?
- mechanizm zmęczenia - neurony z sumieniem (Conscience Learning) uwzględnia częstość wygrywania poszczególnych neuronów pozwalając innym wygrywać konkurencję
- odpowiada chwilowemu zmęczeniu neuronu po wzbudzeniu w sieciach neuronowych biologicznych

Mechanizm zmęczenia z potencjałem

Potencjał p_i neuronu zmieniany

$$p_i(t+1) = \begin{cases} p_i(t) + \frac{1}{n} & \text{dla neuronu zwycięzcy} \\ p_i(t) - p_{min} & \text{dla pozostałych} \end{cases}$$

Neurony dla których $p_i(t) < p_{min}$ nie biorą udziału we współzawodnictwie

$$d(\mathbf{x}, \mathbf{w}_c) = \min\{d(\mathbf{x}, \mathbf{w}_i)\} \quad \text{dla} \quad p_i \geq p_{min}$$

- dla $p_{min} = 0$ nie występuje zmęczenie (standardowy algorytm Kohonena)
- dla $p_{min} = 1$ tylko jeden neuron w danej chwili uczestniczy w uczeniu a reszta odpoczywa
- potencjał p_i ograniczony do 1
- dobre wyniki dla $p_{min} \approx 0.75$

Mechanizm zmęczenia z modyfikacją odległości

Efektywna odległość zależna od częstości zwycięstw N_i

$$\hat{d}(\mathbf{x}, \mathbf{w}_i) = N_i d(\mathbf{x}, \mathbf{w}_i)$$

- kara dodawana przy określaniu zwycięzcy, częściej wygrywające neurony mają większą odległość efektywną
- w trakcie aktualizacji wag używa się oryginalnej odległości
- po kilku początkowych cyklach, wszystkie neurony miały już szansę wygrać, mechanizm się

Gaz neuronowy (Schulten i Martinez 1991)

1. Losowa inicjalizacja N neuronów
2. W kolejnych krokach $t = 1, 2, 3, \dots, T$ wybierz przypadkowy wektor $\mathbf{x}$
3. Zrób ranking neuronów posortowanych względem odległości od $\mathbf{x}$

$$\|\mathbf{w}_i - \mathbf{x}\| \leq \|\mathbf{w}_j - \mathbf{x}\| ; i < j$$

4. Adaptacja

$$\Delta \mathbf{w}_i = \eta_i(t) h_i(t) (\mathbf{v} - \mathbf{w}_i)$$

5. Zmniejsz eksponencjalnie obszar λ i stałą uczenia $\eta(t)$

$$h_i(t) = e^{-\frac{m(i)}{\lambda(t)}}$$

gdzie $m(i)$ to miejsce neuronu w rankingu (dla zwycięzcy $m(i) = 0$)

Promień sąsiedztwa

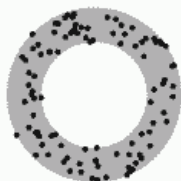
Promień sąsiedztwa maleje w trakcie uczenia od λ_0 do pewnej wartości λ_{min}

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_{min}}{\lambda_0} \right)^{\frac{t}{T}}$$

- dla $\lambda = 0$ uczenie WTA
- dla $\lambda > 0$ aktualizowane są wagi wszystkich neuronów z siłą zależna od pozycji w rankingu

Współczynnik uczenia $\eta(t)$ zazwyczaj maleje wykładniczo

$$\eta(t) = \eta_0 \left(\frac{\eta_{min}}{\eta_0} \right)^{\frac{t}{T}}$$



a) 0 signals



b) 100 signals



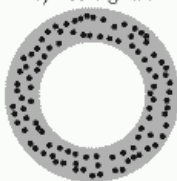
c) 300 signals



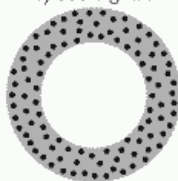
d) 1000 signals



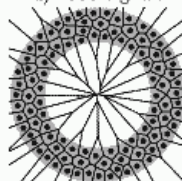
e) 2500 signals



f) 10000 signals



g) 40000 signals



h) Voronoi regions

Growing Cell Structures

Konstruktywny SOM, zbliżony do Growing Neural Gas (Fritzkie)

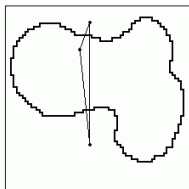
Początkowa topologia: k -wymiarowy sympleks ($k=1, 2, 3$)

Dodaje się nowe neurony i usuwa stare.

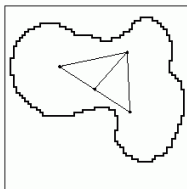
Algorytm SOM, ale bez zmniejszania sąsiedztwa i adaptacji dokonuje się tylko dla zwycięzcy i bezpośrednich sąsiadów.

1. Dla danego $\mathbf{x}$ znajdź zwycięzcę c
2. Popraw wagi zwycięzcy oraz sąsiadów: $\Delta \mathbf{w}_i = \eta_s (\mathbf{x} - \mathbf{w}_i)$
3. Zwiększ licznik częstości τ_c o 1, zmniejsz wszystkie pozostałe o $-\alpha \tau_c$
4. Policz zrenormalizowane częstości $f_i = \frac{\tau_i}{\sum_j \tau_j}$
5. Po ustalonej liczbie epok T znajdź neuron o największej częstości f_i i wstaw pomiędzy ten neuron i najdalszego sąsiada nowy neuron tworząc lokalny sympleks; nowy wektor dla tego neuronu weź z interpolacji użytej pary.

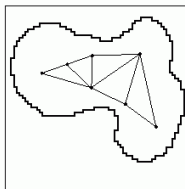
Trening GCS



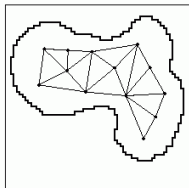
a) 0 signals



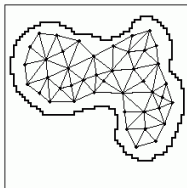
b) 100 signals



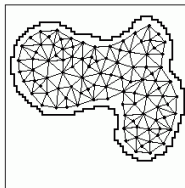
c) 400 signals



d) 1000 signals

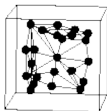
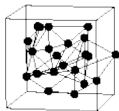


e) 4000 signals

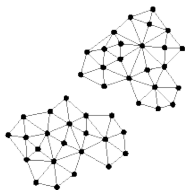


f) 10000 signals

GCS 2 obszary

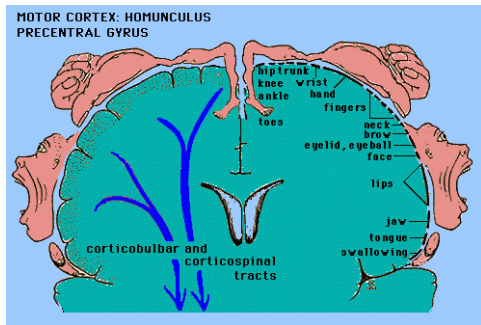
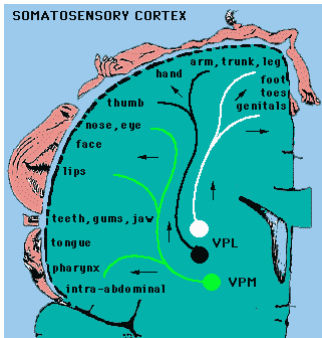


Sytuacja w 3-wym. przestrzeni danych - 2 oddzielone skupienia.



Sieć GCS rosnąca w dwóch wymiarach - odpowiednia topologia.

Mapy czuciowe i motoryczne



Inspiracje biologiczne SOM

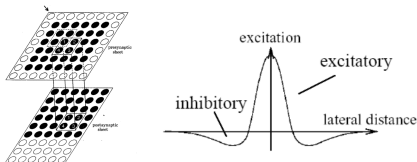
- przetwarzanie różnych danych zmysłowych i motorycznych (wzrok, słuch, dotyk, ...) odbywa się w różnych obszarach kory mózgowej
- neurony o podobnych funkcjach są obok siebie => mapy topograficzne
- istnieje topologiczna zależność w obliczeniach realizowanych przez mózg
- Przykłady
 - mapy somatosensoryczne układu czuciowego,
 - mapy motoryczne kory i mózdzku,
 - mapy tonotopiczne układu słuchowego,
 - mapy orientacji dwuocnej układu wzrokowego,
 - mapy wielomodalne układu orientacji (wzgórki czworacze górne)

Modele samoorganizacji

- Samoorganizująca się mapa cech - SOM lub SOFM (Self-Organized Feature Mapping)
- Połączenia lokalne (sąsiednie neurony): neuron silnie pobudzany przez pobliskie, słabo przez odległe, hamowany przez neurony pośrednie
- von der Malsburg i Willshaw (1976) - model układu wzrokowego
- Amari (1980) - model ciągłej tkanki neuronowej uwzględniający pobudzenie i hamowanie zależne od odległości
- Kohonen (1981) uproszczony model, bez hamowania, dwie fazy uczenia: konkurencja i kooperacja.

Model Willshaw, van der Malsburg (1976)

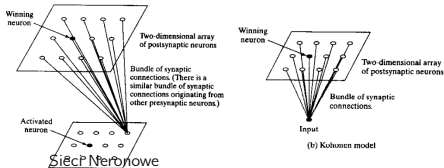
Samoorganizacja w mapowaniu sygnału z siatkówki na korę wzrokową, transformacja 2D -> 2D , brak redukcji wymiaru, połączenia wewnątrzwarstwowe,



- aktywacja typu „Meksykańskiego kapelusza”
 - kooperacja - pobudzenia bliskiego zasięgu
 - konkurencja - hamowanie w dalszych regionach
- siatkówka (wejście) posiada narzuconą organizację topologiczną
- wybór zwycięskiego neuronu w wyniku dynamiki neuronalnej

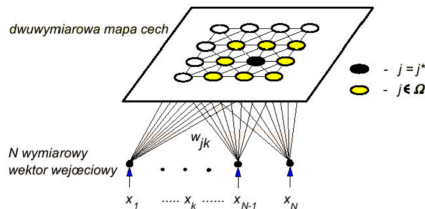
Sieć Kohonena

- Kohonen (1981) – najbardziej popularne uproszczenie, stąd SOM często utożsamiane z „siecią Kohonena”
- dwie fazy
 - konkurencja - najpierw wybierz globalnego zwycięzcę
 - kooperacja - uczenie neuronu zwycięzcy oraz neuronów w sąsiedztwie zwycięzcy
- sieci bez hamowania, brak połączeń wewnątrz warstwy
- wejście jest sygnałem ciągłym, nie posiada narzucoonej struktury topologicznej, dowolna wymiarowość wygnału wejściowego, mapowanie nD do 1D, 2D, 3D



Rys: Marcus Kaiser, Neuroinformatics, lectures

Mapa cech Kohonena



- neurony zachowują porządek topologiczny (np. siatka 2D) tworząc mapę cech
- sąsiadujące neurony powinny reagować na podobne wzorce

Rys: Małgorzata Krętowska, Sztuczne sieci neuronowe

Algorytm SOM

1. Zainicjuj wagi połączeń neuronów ukrytych
2. Dla danego sygnału $\mathbf{x}(t)$ w kroku $t = 1, 2, \dots, T$ znajdź najsilniej reagujący neuron c (np. najbliższy)

$$\|\mathbf{x} - \mathbf{w}_j\| = \sqrt{\sum_i (x_i - w_{ij})^2} \quad c = \arg \min_j \|\mathbf{x} - \mathbf{w}_j\|$$

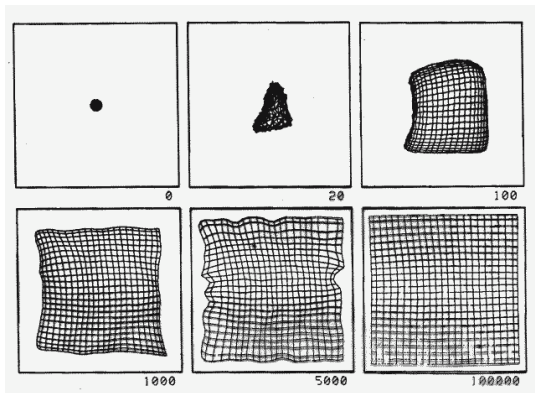
3. Przesuń wagi neuronu c i innych neuronów w sąsiedztwie $O(c)$ w stronę wektora $\mathbf{x}$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) h(\mathbf{r}_i, \mathbf{r}_c, t) [\mathbf{x}(t) - \mathbf{w}_i(t)] \quad \text{dla } i \in O(c)$$

Funkcja $h(\mathbf{r}, \mathbf{r}_c, t)$ określa zasięg sąsiedztwa zwycięzcy

$$h(\mathbf{r}, \mathbf{r}_c, t) = h_0(t) \exp \left(-\frac{\|\mathbf{r} - \mathbf{r}_c\|^2}{2\sigma^2(t)} \right)$$

Przykład: 2D -> 2D

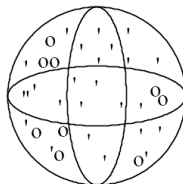


Rozkład jednostajny w kwadracie. SOM uczy się jednorodnego rozkładu. Początkowo wszystkie $\mathbf{w} \approx 0$

Sieć 2D, cęcy 3D

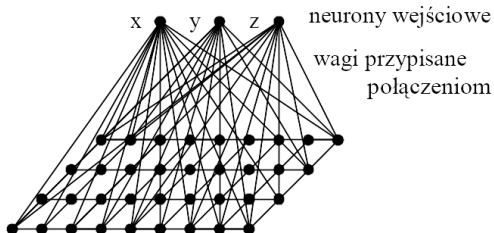
o=dane

' = wagi sieci



przestrzeń cech

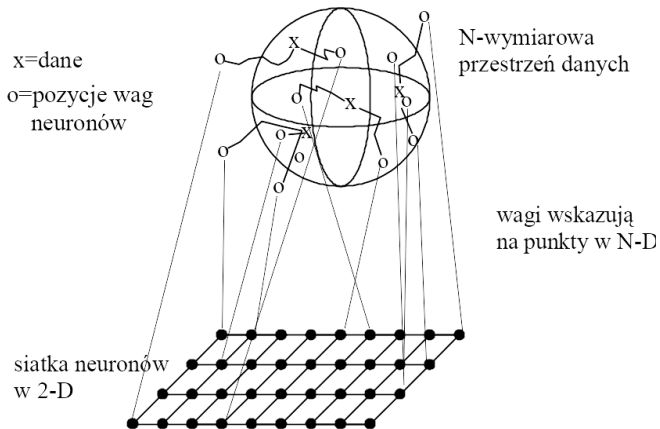
2-D siatka
neuronów



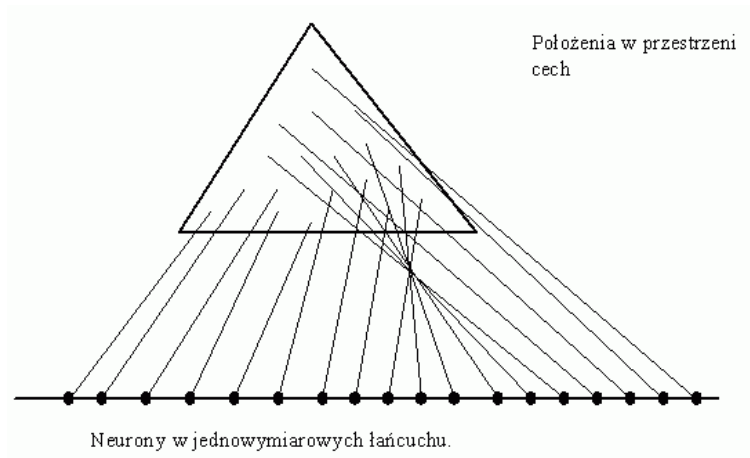
neurony wejściowe

wagi przypisane
połączeniom

Uczenie sieci 2D

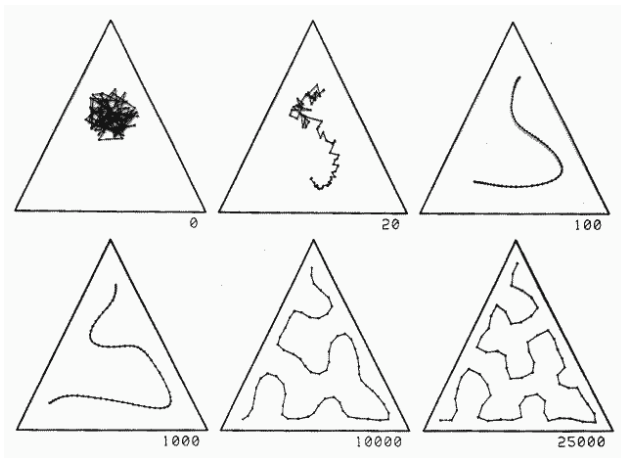


Sieć 2D to 1D

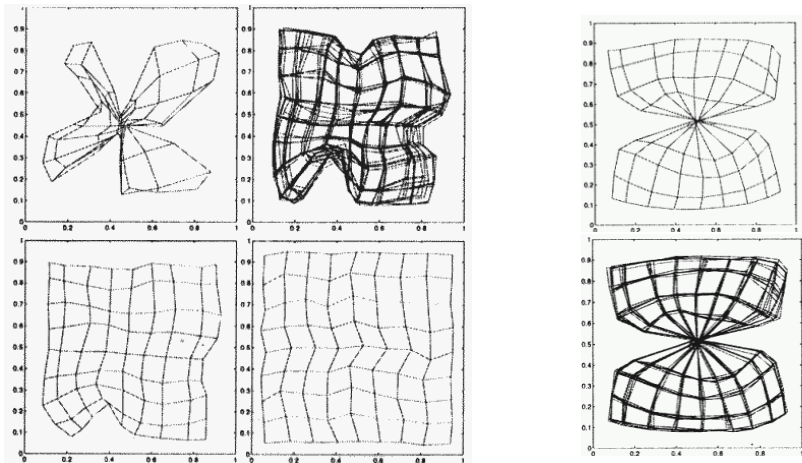


Przykład: fraktalne krzywe Peano

Krzywa Peano (lub Hilberta) - krzywa wypełniająca płaszczyznę



Zniekształcenia



Początkowe zniekształcenia mogą zniknąć lub pozostać.

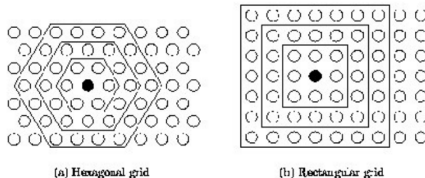
Rozmiar sąsiedztwa

Klasyczny algorytm Kohonena - sąsiedztwo prostokątne

$$h(\mathbf{r}_i, \mathbf{r}, t) = \begin{cases} 1 & \text{dla } d(\mathbf{r}_i, \mathbf{r}) \leq \sigma(t) \\ 0 & \text{dla pozostałych} \end{cases}$$

Promień sąsiedztwa $\sigma(t)$ maleje w czasie treningu, może być miarą odległości lub liczbą neuronów sąsiadujących

- w początkowej fazie uczenia uwzględnia duże sąsiedztwo
- w końcowej fazie tylko neuron zwycięski jest aktualizowany



Funkcja sąsiedztwa SOM

Dla gausowskiej funkcji sąsiedztwa

$$h(|\mathbf{r}_i - \mathbf{r}_j|, t) = \exp\left(-\frac{|\mathbf{r}_i - \mathbf{r}_j|^2}{2\sigma(t)^2}\right)$$

zasięg funkcji maleje w czasie treningu

$$\sigma(t) = \sigma_0 e^{-2\sigma_0 t/t_{\max}}$$

Dla $\sigma = 0$ funkcja sąsiedztwa obejmuje tylko wektor zwycięzcę i SOM sprowadza się do metody kwantyzacji wektorowej (np. on-line k -średnich)

Funkcja błędu SOM

Próba wprowadzenia funkcji błędu (Luttrell; Heskes i Kappen)

Błąd lokalny neuronu i jest sumą po wszystkich neuronach:

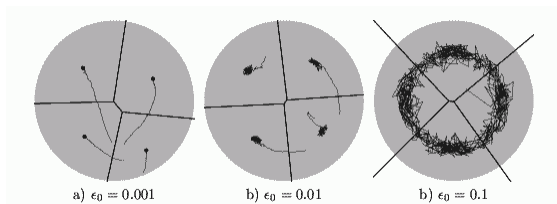
$$E_i(\mathbf{x}; \mathbf{w}, t) = \frac{1}{2} \sum_j h(|\mathbf{r}_i - \mathbf{r}_j|, t) \|\mathbf{x}(t) - \mathbf{w}_j(t)\|^2$$

Neuron-zwycięzca ma najmniejszy błąd lokalny:

$$c = \arg \min_i \sum_j h(|\mathbf{r}_i - \mathbf{r}_j|, t) \|\mathbf{x}(t) - \mathbf{w}_j(t)\|^2$$

Stała uczenia

Duża stała uczenia prowadzi do eksploracji znacznej części przestrzeni.



Stała uczenia maleje w trakcie uczenia

$$\eta(t) = \eta_0 e^{-\frac{t}{t_{max}}}$$

Własności SOM

- Powolna zbieżność algorytmu SOM, wymaga wielu iteracji
- Trudno coś udowodnić o zbieżności lub punktach stacjonarnych. Wyniki analityczne znane są tylko w 1D dla ciągłego czasu: wtedy wartości wag wzdłuż prostej porządkują się.
- W oryginalnym SOM nie ma funkcji błędu, nie ma więc gradientu!
- Sąsiednie neurony kodują sąsiednie obszary, ale sąsiednie obszary mogą być kodowane przez odległe neurony, relacje odległości nie zawsze są zachowane
- „Skręcone” konfiguracje przy zbyt szybkiej redukcji sąsiedztwa
- Jakość klasyfikacji: zwykle niska. Kohonen: SOM służy głównie do wizualizacji ... ale wizualizacja też kiepska, bo brak oceny pozwalającej na redukcję wprowadzanych

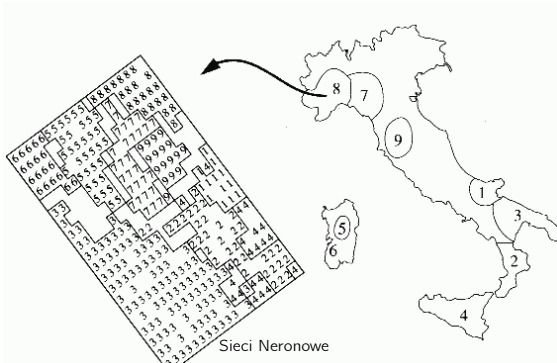
Przykład: Włoska oliwa

Dane: 572 próbki oliwy z 9 prowincji Włoch

Cechy: poziom 8 tłuszczu w każdej próbce.

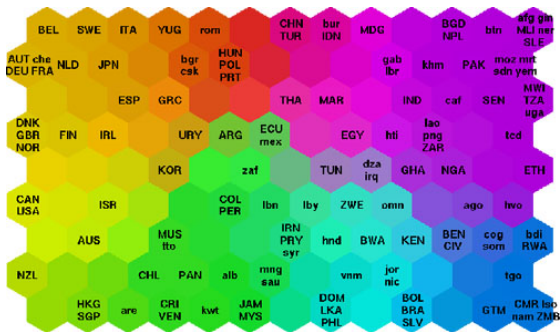
Mapa SOM: 20 x 20, Redukcja 8D => 2D. Dokładność klasyfikacji około 95-97%.

Topograficzne relacje zostały zachowane.



Przykład: Wodr powerty map

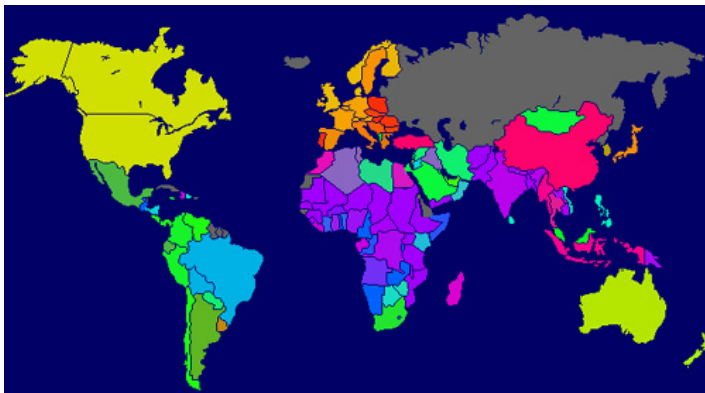
Dane: WorldBank data 1992 rok, 39 współczynników opisujących jakość życia w kraju (poziom służby zdrowia, edukacji, etc.)



Kraje o podobnej wartości poziomemu życia pobudzają bliskie grupy neuronów, od krajów o najwyższym wsp. jakości życia (żółty) do najniższego (niebieski)

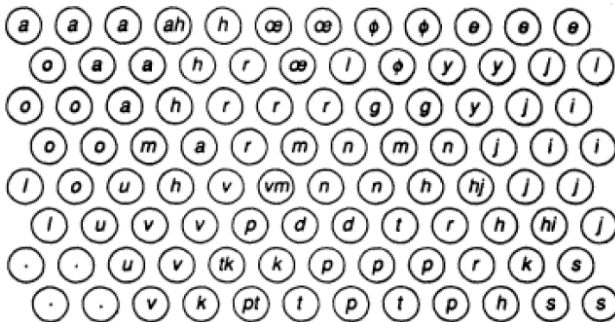
Przykład: Wodr powerty map

Po naniesieniu kolorów na mapę świata



Klawiatura fonetyczna

Dane: 10 ms próbki FFT mowy



Mapa fonetyczna dla języka Fińskiego, Kohonen 1980

Demos

- Wizualizacje do wykładu M. Czoków, J. Piersa:  Sześcian,  Sześcian z funkcją gaussowską,  Kula,  Kula z funkcją gaussowską
-  SOM 2D dla pogrupowanych danych
- Self-organizing Maps: PyMVPA kolory na mapie 2D
-  Demo GNG
-  Klasyfikacja kolorów
-  GeoSOM Environmental modelling

Przykłady zastosowania SOM

Helsinki University of Technology web site

<http://www.cis.hut.fi/research/refs/> has a list (2010) of
> 7700 papers on SOM and its applications !

- Badania mózgu: modelowanie topograficznych map w różnych obszarach kory (motoryzna, sensoryczna, wizyjna)
- Robotyka i AI: analiza danych z czujników, sterowanie ruchami robota, mapy orientacji przestrzennej
- Kategoryzacja dokumentów
- Analiza skupień genów, białek, cząteczek chemicznych, fonemów, dźwięków zwierząt, obiektów astronomicznych, dane ekonomiczne i finansowe
- Kompresja danych (obraz, dźwięk), filtrowanie informacji
- Medical and technical diagnostics.

- Analiza języka naturalnego: linguistic analysis, parsing, learning languages, hyphenation patterns
- Problemy optymalizacyjne: configuration of telephone connections, VLSI design, time series prediction, scheduling algorithms.
- Przetwarzanie sygnałów: adaptive filters, real-time signal analysis, radar, sonar seismic, USG, EKG, EEG and other medical signals ...
- Rozpoznawanie i analiza obrazów: segmentation, object recognition, texture recognition ...
- Content-based retrieval: examples of WebSOM, PicSom – similarity based image retrieval, RGB and textures.
- Viscovery SOMine, komercyjny program do wizualizacji, eksploracji, klasyfikacji, prognozowania oparty na SOM.