

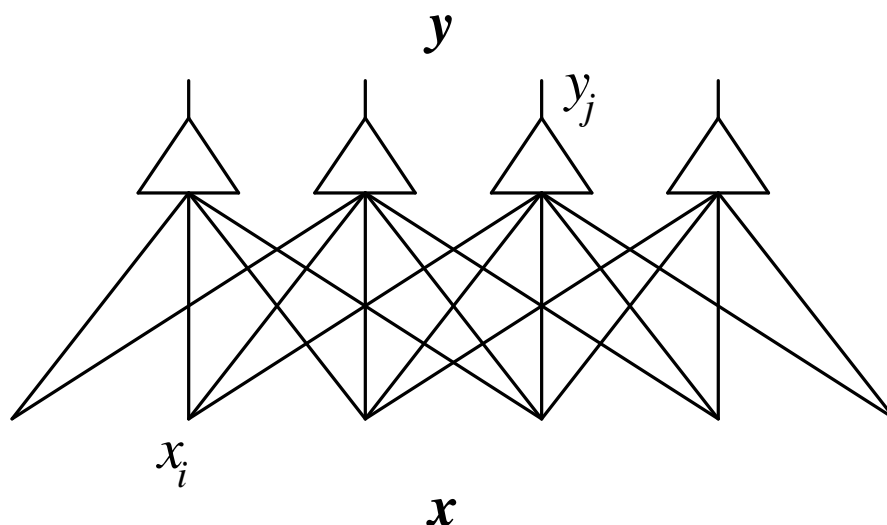
Wykład 5: Uczenie przez współzawodnictwo

W sieci uczoney w sposób *konkurencyjny* (przez konkurencję, przez *współzawodnictwo*) proces uczenia ma charakter **wybiórczy**, w odróżnieniu od większości innych sieci, uczonych z reguły w całości. Po prezentacji i propagacji kolejnego przykładu w sieci aktywny jest tylko jeden neuron (lub niewielka ich grupa), pozostałe, których pobudzenia były zbyt niskie, "przegrywają" we współzawodnictwie i nie są w ogóle brane pod uwagę w bieżącym kroku uczenia.

Uczenie przebiega według zasady "**zwycięzca bierze wszystko**" (*Winner Takes All*): modyfikacja wag przeprowadzana jest jedynie dla neuronu (neuronów) najbardziej aktywnego. "Przegran" nie podlegają żadnym modyfikacjom, a czasami nawet symuluje się w nich proces "zapominania" lub "oduczania".

Cel: kwantowanie wektorowe, czyli nienadzorowane grupowanie wektorów wejściowych stosownie do ich wzajemnego podobieństwa (nienadzorowana kategoryzacja, *clustering*).

Najprostsza sieć realizująca uczenie z konkurencją: pojedyncza warstwa liniowych neuronów.



gdzie:
$$y_j = \mathbf{w}_j \mathbf{x} = \sum_i w_{ij} x_i$$

\mathbf{w}_j - wektor wag j -tego neuronu

Zwycięzca:

$$win = \max_j \arg y_j$$

Przy znormalizowaniu wektorów wag wszystkich neuronów w sieci:

$$\forall j \quad |\mathbf{w}_{win} - \mathbf{x}| \leq |\mathbf{w}_j - \mathbf{x}| \quad (5.1)$$

Możliwe realizacje:

- poszukiwanie zwycięzcy przez przeglądanie neuronów;
- wykorzystanie zjawiska **hamowania obocznego**: każdy neuron hamuje wszystkie pozostałe, a pobudza samego siebie (wymagana nieliniowa funkcja aktywacji, szczegóły w [Grossberg 1976, 1980]).

Jak nagradzać (czyli uczyć) zwycięzcę ?

Początkowa konfiguracja wag - losowa.

Uczenie zwycięskiego neuronu polega na zbliżeniu ("przekręceniu") jego wektora wag w kierunku wektora wejść, podobnie jak w regule delta, np.:

$$\Delta \mathbf{w}_{win} = \eta \mathbf{x}$$

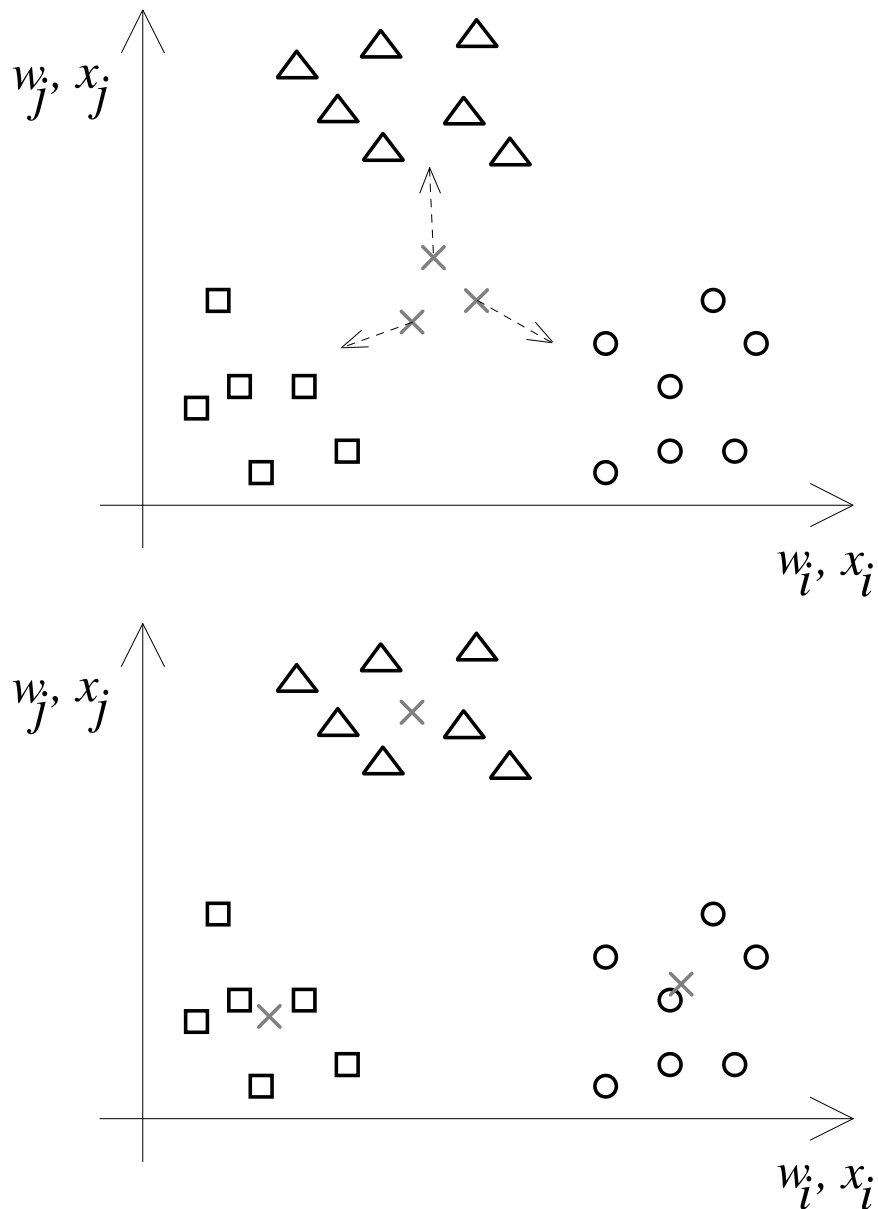
Wadą tej formuły jest to, że prowadzi ona do nieograniczonego wzrostu wielkości wag. Da się tego uniknąć normalizując wektor wag po każdej modyfikacji lub stosując bardziej wyrafinowaną formułę tak, aby zachowany był warunek $\sum_i w_{ij} = 1$ lub $\sum_i w_{ij}^2 = 1$.

Najczęściej do uczenia przez współzawodnictwo używa się reguły w postaci:

$$\Delta \mathbf{w}_{win} = \eta (\mathbf{x} - \mathbf{w}_{win}) \quad (5.2)$$

stosując normalizację zarówno wektora wag, jak i wektora wejść.

Przykładowy przebieg procesu uczenia:



(W pewnym sensie sieć znajduje tzw. *obiekty modalne skupień*).

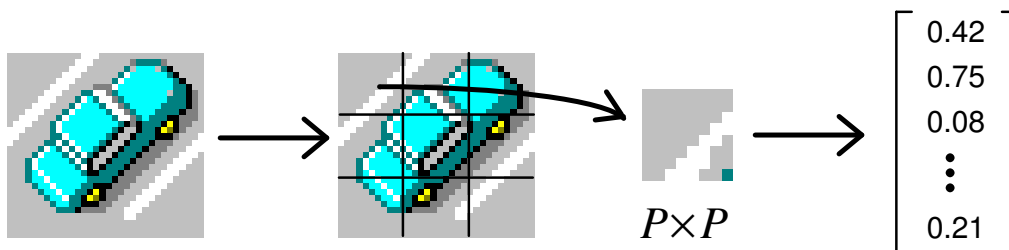
Wady sieci uczonych przez konkurencję:

- "rozrzutne" kodowanie klas (kod 1 z N);
- mniejsza w porównaniu z innymi typami sieci odporność na uszkodzenia;
- brak możliwości tworzenia hierarchii kategorii;
- problem tzw. **jednostek martwych**, gdy neuronów jest więcej niż skupień.

Przykłady zastosowań

- Przybliżona bisekcja grafu: podział grafu o N wierzchołkach na dwa równoliczne podgrafy z minimalną liczbą krawędzi pomiędzy nimi.
Architektura sieci: N wejść reprezentujących wierzchołki grafu, 2 wyjścia odpowiadające podgrafom.
Reprezentacja danych: Krawędź łącząca wierzchołki i -ty z j -tym reprezentowana jest jako wektor binarny z wartościami 1 na pozycjach i -tej i j -tej i pozostałymi pozycjami równymi 0.
Zbiór uczący: krawędzie grafu.
- Kompresja danych przez kwantowanie wektorowe. Uczenie przez współzawodnictwo służy tu do znajdowania tzw. *książki kodowej*, tj. zbioru tzw. *wektorów prototypowych*.

Przykład: **kompresja obrazu** (w celach archiwizacji lub transmisji).



Uczenie przebiega na zbiorze wielu obrazów. Każdy obraz dzielony jest na regularne pola o wielkości $P \times P$ pixeli; każde z nich stanowi wektor wejściowy dla sieci. Sieć w trakcie uczenia, dysponując ograniczoną ilością $N < P \times P$ neuronów, stara się dokonać możliwie dobrej kategoryzacji, przyporządkowując podobnym wektorom (fragmentom obrazu) te same neurony (zwycięzców).

Testowanie: obraz testowy podawany jest pole po polu na wejścia sieci; dla każdego pola zapamiętujemy indeks zwycięzcy (sieć pełni tu rolę tzw. książki kodowej). Zamiast archiwizować/transmitować poszczególne punkty obrazu, archiwizujemy/przesyłamy jedynie indeksy. Przy odtwarzaniu obrazu posługujemy się tą samą nauczoną siecią i indeksami neuronów, odtwarzając na tej podstawie kolejne fragmenty obrazu.

Przykład: Niech $P=8$, $N=256$, liczba stopni szarości (lub kolorów) $G=256$. Wówczas stopień kompresji wynosi:

$$\frac{P \times P \times \text{sizeof}(G)}{\text{sizeof}(N)} = \frac{8 \times 8 \times 1\text{byte}}{1\text{byte}} = 64$$

Nadzorowane uczenie konkurencyjne na przykładzie LVQ

Learning Vector Quantization, Adaptacyjne Kwantowanie Wektorowe
[Kohonen 1989]

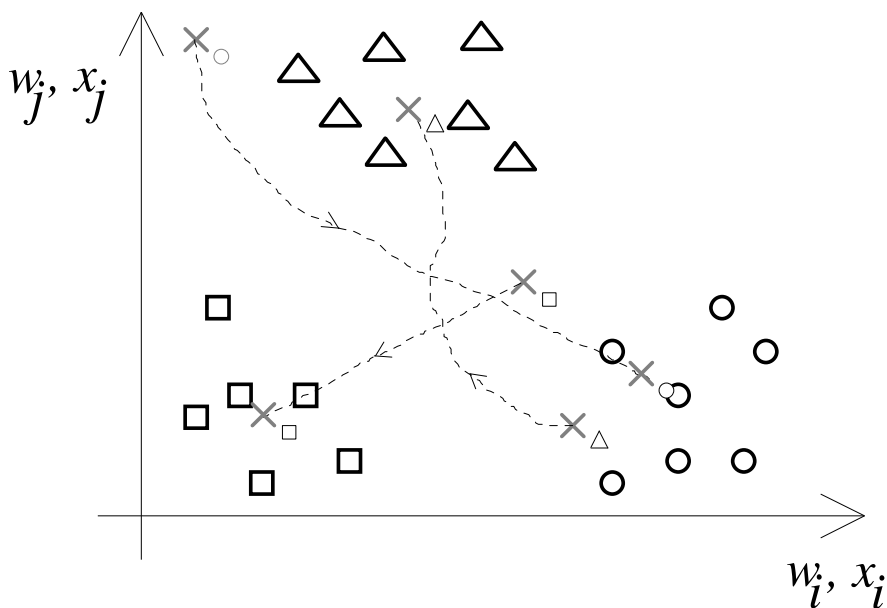
Uczenie nadzorowane - przynależność przykładów do klas jest znana.

Architektura i działanie sieci - jak w nienadzorowanym uczeniu przez konkurencję z tą różnicą, że neurony przypisane są *a priori* do określonych klas k . Istotnej modyfikacji w stosunku do reguły standardowej (5.2) podlega reguła uczenia:

$$\Delta \mathbf{w}_{win}^{(k)} = K(k, k') \eta (\mathbf{x}^{(k')} - \mathbf{w}_{win}^{(k)}) \quad K(k, k') = \begin{cases} +1, & \text{gdy } k = k' \\ -1, & \text{gdy } k \neq k' \end{cases}$$

gdzie: K - funkcja zgodności klasy, k - klasa, do której przypisany jest neuron, k' - klasa, do której należy przykład.

Czyli: Jeżeli zwyciężski neuron reprezentuje tę samą klasę co prezentowany przykład, wektor wag jest zbliżany do wektora przykładów; w przeciwnym razie jest od niego oddalany.



- Czasami dobrze jest tak skonstruować sieć, aby do każdej klasy przypisanych było po kilka neuronów.
- Ulepszona wersja, LVQ2 [Kohonen i in. 1989], osiąga wysoką trafność klasyfikowania i daje bardzo dobre rezultaty w zastosowaniach praktycznych (np. przetwarzaniu i rozpoznawaniu mowy).

SOM - sieć odwzorowania cech istotnych uczona przez współzawodnictwo

Self Organizing Map, sieć Odwzorowania Cech Istotnych [Kohonen 1982]

Idea: Do zwykłej sieci uczonej przez współzawodnictwo dodajmy aspekt topologii - istotne staje się położenie wygrywającego neuronu w warstwie wyjściowej.

Cel: Otrzymanie sieci, w której podobieństwo wzorców podawanych na jej wejście odzwierciedla się w "fizycznej" odległości zwycięzców w warstwie wyjściowej. Im dwa wzorce są bardziej do siebie podobne, tym bliższe są odpowiadające im wygrywające neurony, i *vice versa*.

Topologia i uczenie:

Stosujemy zwykłą sieć uczenia przez konkurencję, z tradycyjnym kryterium znajdowania zwycięzcy (wzór 5.1). Neurony aranżujemy w proste układy geometryczne, najczęściej jednowymiarowe (prosta) lub dwuwymiarowe (tablica prostokątna). Definiujemy *funkcję sąsiedztwa* $\Lambda(i, j)$, której wartość dla każdej pary neuronów (i, j) odzwierciedla geometrię układu. $\Lambda(i, i) = 1$, a gdy odległość pomiędzy neuronami i i j rośnie, $\Lambda(i, j)$ zanika do zera. Zazwyczaj przyjmuje się:

$$\Lambda(i, j) = e^{-\frac{euklid(i, j)}{\beta}}$$

Przy pomocy tak zdefiniowanej funkcji sąsiedztwa modyfikuje się regułę uczenia (5.2)

$$\Delta \mathbf{w}_{win} = \eta (\mathbf{x} - \mathbf{w}_{win})$$

w sposób następujący:

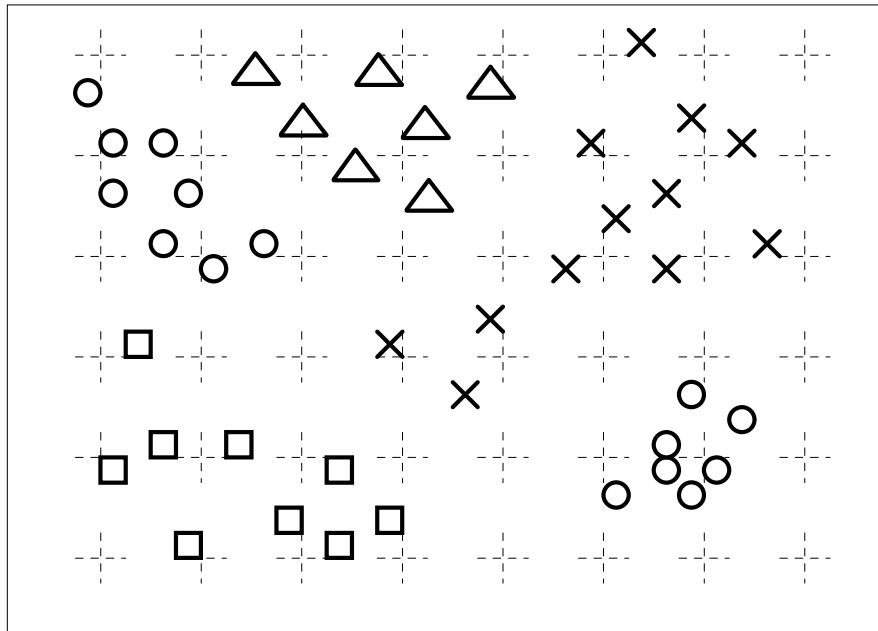
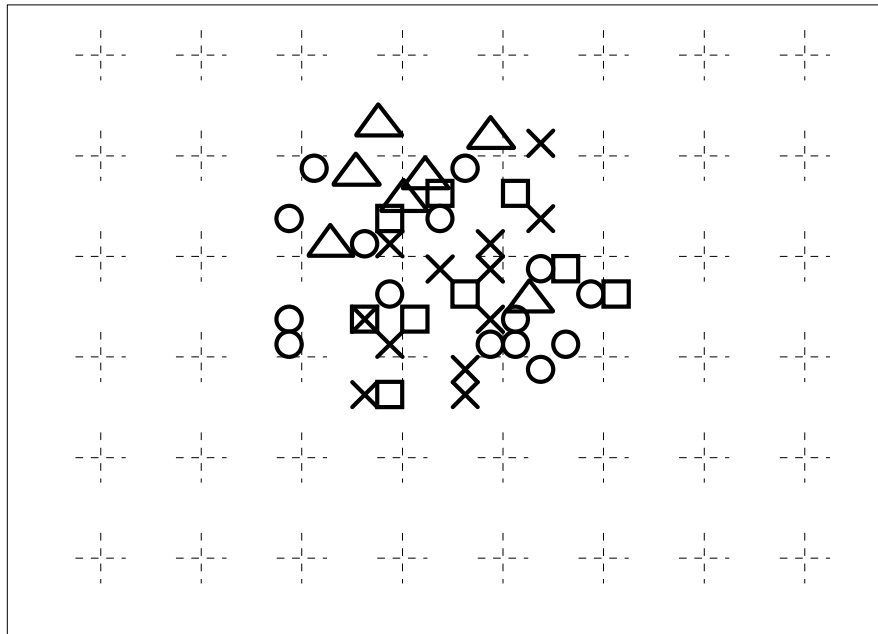
$$\forall j \quad \Delta \mathbf{w}_j = \eta \Lambda(j, j_{win}) (\mathbf{x} - \mathbf{w}_j)$$

gdzie: j - indeks neuronu

j_{win} - indeks neuronu wygrywającego

W rezultacie największych modyfikacji wag doświadcza zwycięzca i jego "sąsiedzi". Im dalej od neuronu zwycięskiego, tym mniejsza intensywność procesu uczenia.

Przykładowy przebieg uczenia (początkowa konfiguracja wag losowa):




Działanie sieci SOM polega na (ciągłym) odwzorowywaniu przestrzeni wielowymiarowej w przestrzeń o (z reguły) mniejszej liczbie wymiarów (zwykle 1, 2 lub 3). Sieć "rozciąga", "ściska", "zgina" i "związa" na różne sposoby przestrzeń przykładów wejściowych tak, aby przy możliwie wiernym zachowaniu topologii mapować ją na przestrzeń neuronów wyjściowych.

Uwagi:

- Opisywane podejście reprezentuje uczenie nienadzorowane: wstępny podział przykładów na klasy nie jest wymagany.
- W praktyce często stopniowo zmniejsza się η i β podczas uczenia: zmniejszanie η stopniowo spowalnia uczenie, zaś dekrementacja β za-węża sąsiedztwo.
- W oryginalnym podejściu wygrywa zawsze jeden neuron, tak więc macierz wyjść zawiera zawsze jedną jedynkę, przy pozostałych wartościach równych zeru. Dla pewnych zastosowań taka informacja ma zbyt przybliżony ("zgrubny") charakter, stąd też często próbuje się ekstrahować z sieci bardziej precyzyjną informację. Podczas testowania, po propagacji wektora wejściowego i znalezieniu zwycięzcy przegląda się jego otoczenie w poszukiwaniu innych neuronów, które co prawda nie zwyciężyły, ale "są w czołówce" (złagodzenie warunku zwycięstwa). Po wyznaczeniu "czołówki" można dokonać jej uśrednienia (np. przez znalezienie jej środka ciężkości) i tak otrzymane współrzędne potraktować jako ostateczną odpowiedź sieci.
- Czasami przykłady w wyniku uczenia rozpraszają się po peryferiach mapy, tworząc zgrupowania na jej obrzeżach i w jej narożnikach stosownie do podobieństwa "klas", do których należą (niekoniecznie *klas* w sensie ML). Dzieje się tak, ponieważ skrajna lokalizacja na mapie zapewnia z reguły optymalną (tj. maksymalną) odległość od przykładów innych klas. Zjawisku temu można zapobiec "sklejając" ze sobą przeciwległe krawędzie mapy w taki sposób, aby otrzymać powierzchnię bez brzegu. Daje to sieci dodatkowy stopień swobody i pozwala z reguły na dokładniejszą analizę struktury zbioru przykładów.
- W ogólności neurony mogą tworzyć układy o różnej architekturze (często spotyka się np. układ typu "plaster miodu"). Podobnie manipulować można definicją sąsiedztwa.

Przykłady zastosowań:

- Wizualizacja przestrzeni przykładów opisanych wieloma atrybutami. Z otrzymanej mapy można ze znaczną dozą pewności wnioskować o takich aspektach zbioru danych jak: podobieństwo przykładów pochodzących z różnych klas, zwartość zbioru przykładów reprezentujących daną klasę, obecność "podklas" (kiedy przykłady danej klasy występują w więcej niż jednym skupieniu, na rysunku z poprzedniej strony przykłady oznaczone )
- Sterowanie ramieniem robota
- Kwantowanie wektorowe

Rozwiązywanie problemu komiwojażera (metoda siatki elastycznej)

Dane: M "miast": A, B, C, \dots o znanych położeniach (współrzędnych):
 $(x_A, y_A), (x_B, y_B), (x_C, y_C) \dots$

Cel: Minimalizacja funkcji celu odzwierciedlającej długość zamkniętej marszruty:

$$H = \sum_{m=1}^M d_{X_m X_{m+1}} \quad (X_{M+1} = X_1)$$

gdzie: d_{XY} - odległość pomiędzy miastami X i Y

Problem ten da się potraktować jako zadanie znajdowania odwzorowania przestrzeni dwuwymiarowej - płaszczyzny (mapa z naniesionymi "miastami") na przestrzeń jednowymiarową - linię (marszrutę komiwojażera).

Architektura sieci

- Dwa wejścia, odpowiadające współrzędnym (x, y) miast;
- M neuronów reprezentujących miasta, tworzących jednowymiarową (liniową) warstwę.

Konfiguracja początkowa

Wagi kolejnych neuronów odpowiadają kolejnym punktom na zamkniętej marszrucie (np. na okręgu).

Podczas **uczenia** podawane na wejścia sieci miasta "przeciągają" na swoją stronę poszczególne neurony. Marszruta zachowuje się jak elastyczna gumka, którą sieć stara się "rozpiąć" pomiędzy miastami. W konfiguracji końcowej wagi neuronów odpowiadają (w przybliżeniu) współrzędnym miast, a kolejność neuronów w warstwie wyjściowej reprezentuje rozwiązanie: kolejność odwiedzin miast.

Uwagi:

- Lepiej jest mieć więcej neuronów niż miast (stosuje się algorytmy dodające i usuwające jednostki w razie potrzeby).
- Wada: brak funkcji kosztu, pokazującej, w jakim stopniu otrzymane rozwiązanie pośrednie jest optymalne (ponieważ rozwiązania robocze, wypracowywane przez sieć podczas uczenia, są z reguły niepoprawne).

Zastosowanie SOM do analizy mowy

The "neural" phonetic typewriter [Kohonen 1988]

Dane wejściowe: Sygnał mowy (pojedyncze słowa).

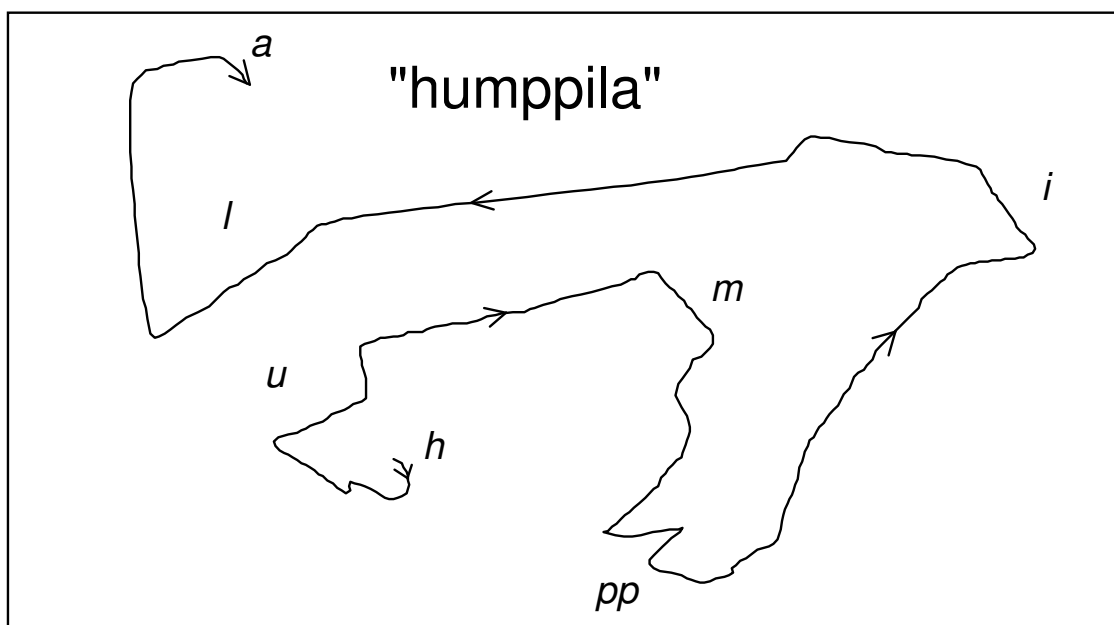
Wstępna obróbka danych: 1) 12-bitowy przetwornik analogowo-cyfrowy próbkuje sygnał wejściowy co ~10ms; 2) 256-punktowa szybka analiza Fouriera (FFT); 3) wyjście: 15-punktowe spektrum sygnału wejściowego.

Sieć: SOM: 15 wejść, warstwa wyjściowa: $8 \times 12 = 96$ neuronów.

Uczenie: Sygnał mowy (pojedyncze słowa).

Kalibracja: Po nauczaniu sieci podawano na jej wejścia "wzorcowe" fonemy, opatrując zwycięzców odpowiednimi etykietami. Prawie każdej klasie fonemów odpowiada jeden zwycięzca lub grupa blisko położonych zwycięzców.

Testowanie: Podczas wypowiedzania słowa notowane jest położenie zwycięzcy, które zmieniając się zakreśla pewną marszrutę. Otrzymana w ten sposób transkrypcja fonetyczna słowa może być np. podana na wejście systemu regułowego, który na tej podstawie będzie generował odpowiednie słowo.



Uwagi:

- system działa rzeczywiście jak automatyczna maszyna do pisania: dokonuje detekcji fonemów nie "rozumiejąc" słów,
- trafność rozpoznawania fonemów: 92-97%,
- system działa w czasie rzeczywistym,
- odseparowane słowa a ciągły sygnał mowy,
- ciekawostka: neurony w warstwie wyjściowej ułożone w "plaster miodu".