Elementy Sztucznej Inteligencji

Sztuczne sieci neuronowe cz. 2

Plan wykładu

- Uczenie bez nauczyciela (nienadzorowane).
  - Sieci Kohonena (konkurencyjna)
- Sieć ze sprzężeniem zwrotnym – Hopfielda.

Cechy uczenia nienadzorowanego

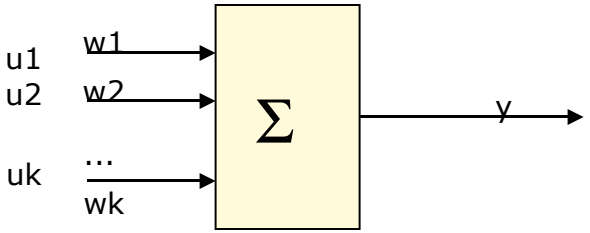
- Nie znana jest odpowiedź Z na zadany wektor wejściowy U. Korekta wag nie jest zdeterminowana błędem sieci (nie można go policzyć nie znając Z).
- Uczenie nienadzorowane (bez nauczyciela) odwzorowuje procesy uczenia się bez instruktażu.
- Czego zatem uczy się sieć? **Struktury danych**. W danych mogą być zawarte pewne cechy, które można odkryć.

Porównanie nadzorowanego i nienadzorowanego

- Uczenie z nauczycielem:
  - prezentuje się wzorce i odpowiedzi
  - cel SSN: znalezienie zależności pomiędzy danymi wejściowymi i odpowiedziami
- Uczenie bez nadzoru:
  - prezentuje się tylko wzorce
  - cle SSN: wykrycie cech zawartych w sposób ukryty w danych
  - sieć próbuje nauczyć się struktury danych

Architektura sieci

- Architektura sieci: sieć jednowarstwowa z liniową funkcją aktywacji (mogą być też signum, progowa).



$$y = \sum_{i=1}^{n} w_i u_i$$

Reguła Instar (poprzednik rywalizacji)

- Wagi (w procesie uczenia) są zmieniane tylko, gdy wyjście jest aktywne.

$$\Delta w_{ij} = \eta \cdot u_i \cdot y_j - \gamma \cdot w_{ij} \cdot y_i$$

$$\eta = \gamma \quad \Rightarrow \quad \Delta w_{ij} = \eta \cdot y_i \cdot \left(u_j - w_{ij} \right)$$

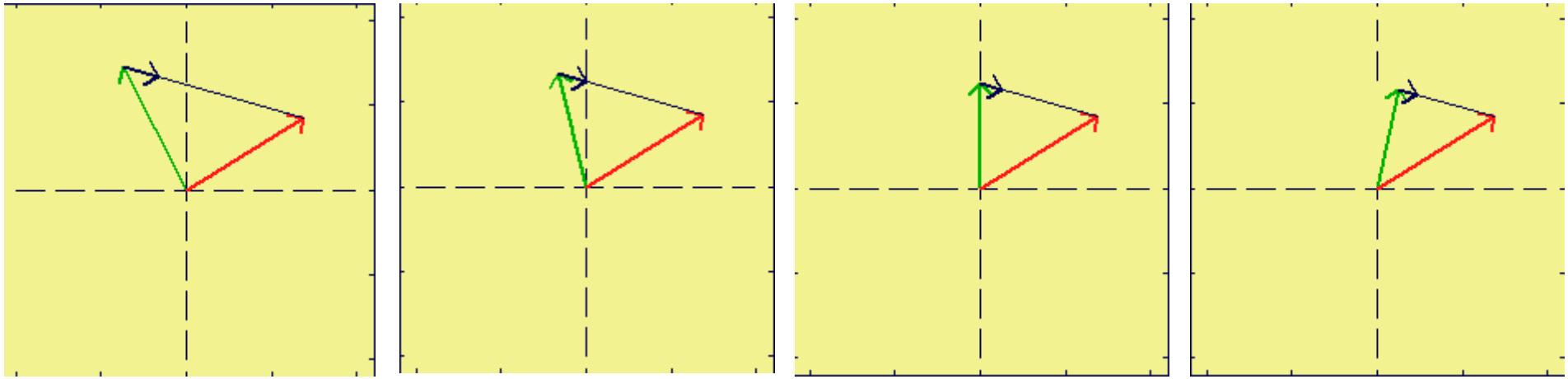
$$y_i = 1 \quad \Rightarrow \quad \Delta w_{ij} = \eta \cdot \left(u_j - w_{ij} \right)$$

$$\Rightarrow \quad w_{ij}(t) = (1-\eta) \cdot w_{ij}(t-1) + \eta \cdot u_j$$

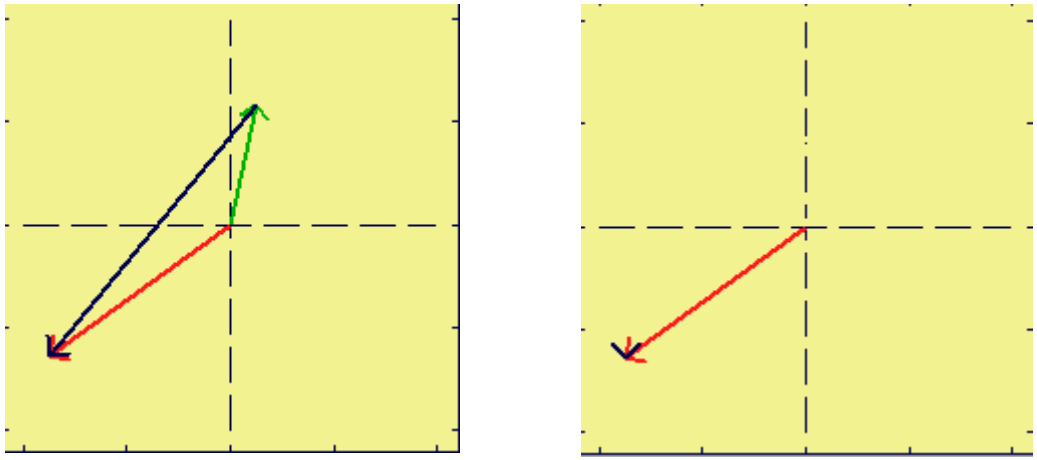
Instar

- Wektor wag przemieszcza się w kierunku wektora wejść.

Interpretacja geometryczna



$\eta = 0.2$



$\eta = 1$

Działanie reguły Instar

- Koncepcja polega na wybraniu neuronu i narzuceniu mu strategii uczenia, by zapamiętał i potrafił rozpoznać sygnał wejściowy *U*.
- Inne neurony w sieci będą wówczas bezczynne.
- Stosowany, gdy trzeba nauczyć sieć rozpoznawania określonego sygnału *U*.

Uczenie z rywalizacją

- Reguła Kohonena wywodzi się z instar.

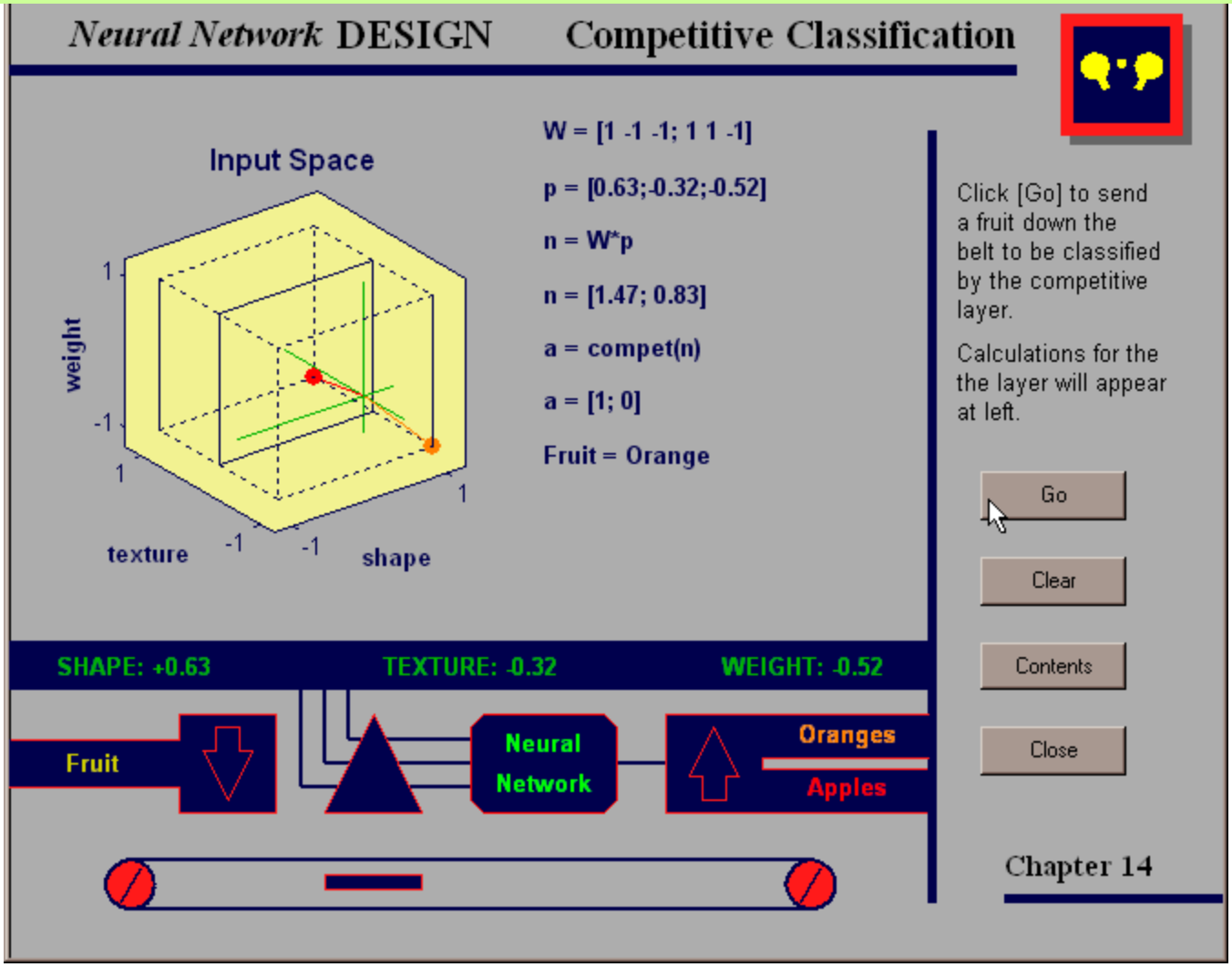
$$w_{ij}(t) = w_{ij}(t-1) + \eta \cdot (u_j - w_{ij}(t-1))$$

- Zakłada się jednak, że wektor wejść U musi być znormalizowany ||U||=1.
- Nie wybiera się też arbitralnie neuronu do uczenia, tylko uczy się ten, który „zwycięża” – ma największy sygnał wyjściowy.
- Tylko neuron „zwycięzca” (Winner-Takes-All) jest uczony i zmieniane są jego wagi, tak by najlepiej dopasował się do wektora wejściowego, którego się uczy.
- Inne wektory wag nie zmieniają się.

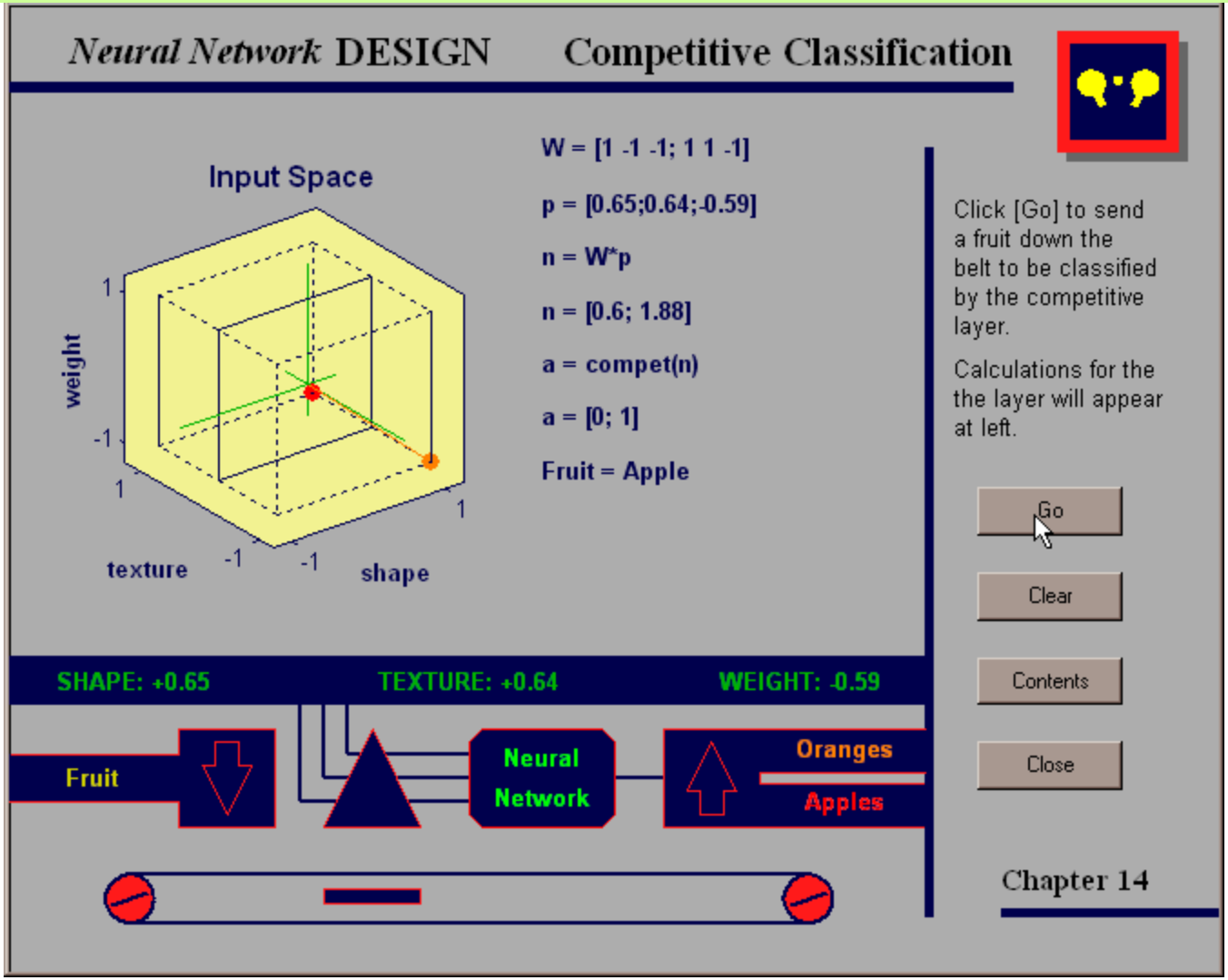
Przykład klasyfikacji z regułą Kohonena

- Rozpoznajemy jabłka i pomarańcze na podstawie 3 cech: waga, faktura i kształt.
- Sieć składa się z dwóch neuronów. Sygnał wejściowy jest przekazywany do obu neuronów. Pierwszy neuron uaktywni się, gdy rozpozna pomarańczę, a drugi , gdy jabłko.
- Sieć nauczono rozpoznawania za pomocą reguły Kohonena.

Rozpoznanie pomarańczy



Rozpoznanie jabłka



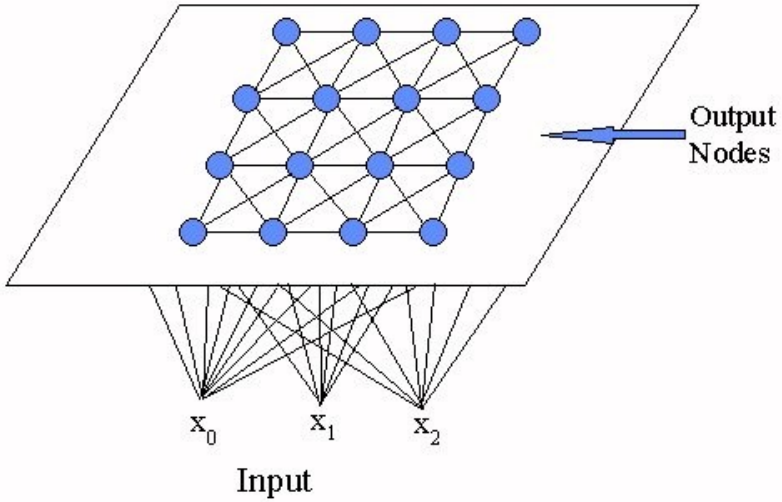
Samoorganizująca się mapa Kohonena

- Dodaje się do sieci z rywalizacją element sąsiedztwa.
- Oprócz najlepszego neuronu wygrywają wszystkie leżące w jego sąsiedztwie (Winner-Takes-Most) (które może być określone np.. jako odległość Manhattan).
- Pojęcie sąsiedztwa i jego promień mogą być różne.

Graficzna prezentacja dwuwymiarowej SOM

- Każdy neuron z warstwy wejściowej połączony jest z każdym neuronem z mapy. Wagi propagują wartości wejść do mapy neuronów.
- Wszystkie neurony w mapie są połączone pomiędzy sobą. Połączenia te wpływają na neurony w pewnym założonym obszarze aktywności o największej aktywacji.
- Wielkość zasięgu jest obliczana funkcją gaussa

  x_c=pozycja neuronu zwycięzcy
  x_i=pozycja innego neuronu
  sig=promień sąsiedztwa

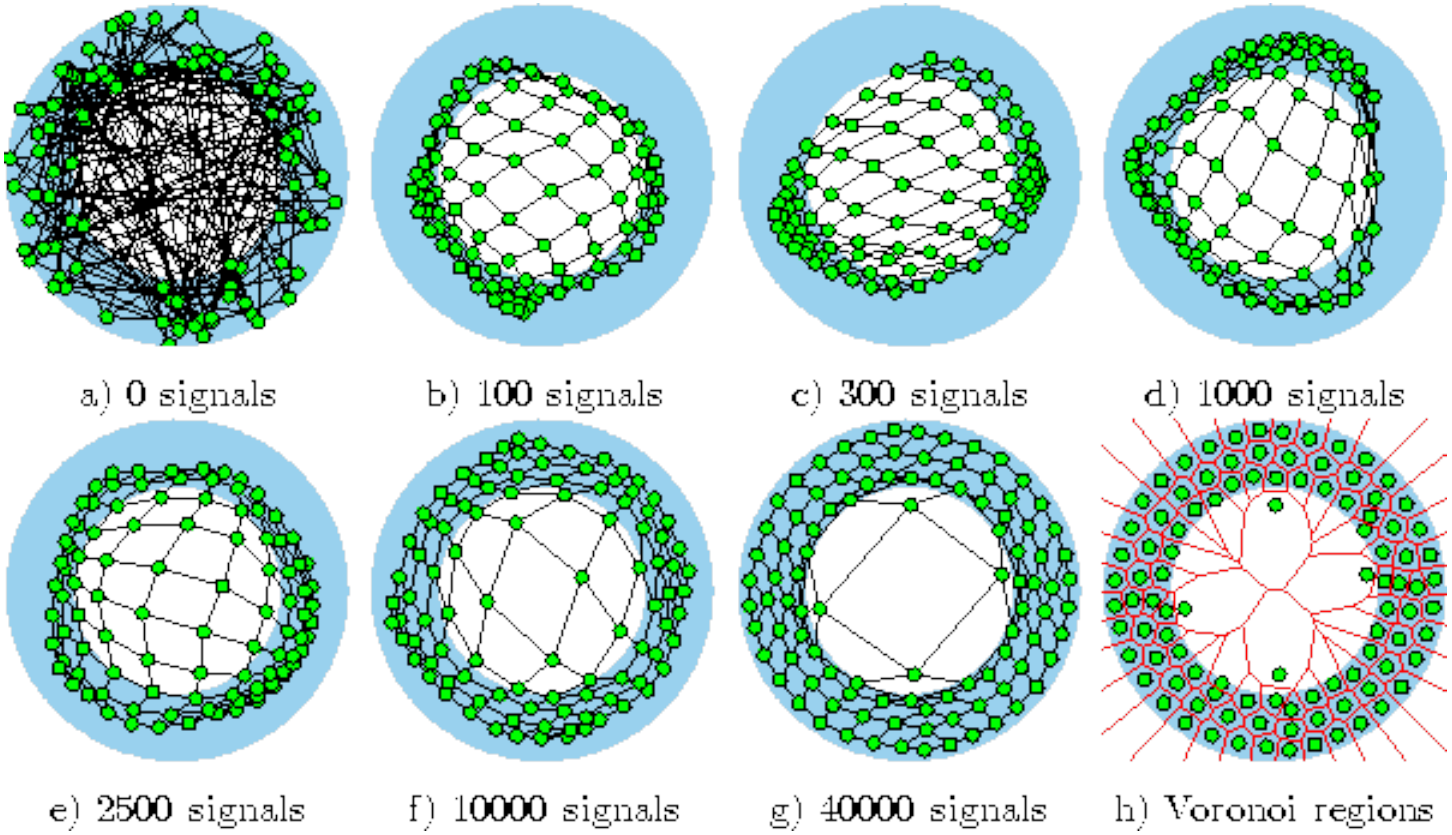


$$\text{zasieg}_{c_i} = e^{\frac{-|x_c - x_i|^2}{2\,sig}}$$

Algorytm uczenia

- Wybierz obiekt ze zbioru uczącego.
- Znajdź węzeł najbliższy wybranej danej (tj. odległość pomiędzy w_{ij} i danymi u_i jest minimalna).
- Zmień wektor wag wybranego węzła i jego sąsiadów zgodnie z regułą.
- Powtarzaj ustaloną ilość razy.
- Zazwyczaj stosuje się zmienny (malejący) współczynnik uczenia.
- Wielkość sąsiedztwa też może być zmienna i maleje z czasem.

Przykład dopasowania do danych rozłożonych w kształt pierścienia



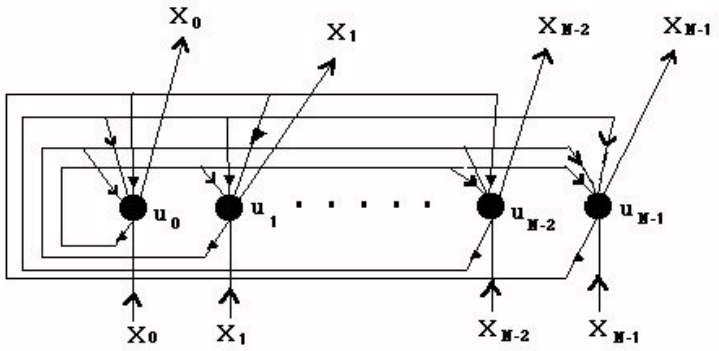
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node22.html

Zastosowania sieci Kohonena

- Eksploracyjna analiza danych – sieć uczy się rozpoznawania skupień występujących w danych wejściowych i może nauczyć się kojarzenie podobnych klas.
- Wykrywanie nowości: Jeżeli podana dana nie pasuje do nauczonych wzorców, oznacza to, że zaprezentowano nowe dane, nie istniejące w żadnej ze zbudowanych klas.
- Graficzna reprezentacja zależności pomiędzy danymi.

Sieci Hopfielda

- Przykład sieci ze sprzężeniem zwrotnym.
- Brak podziału na warstwy.
- W pierwowzorze każdy neuron połączony jest z każdym, a sygnały wyjściowe przekierowane są jako wejścia.



Pamięć

- Sieć Hopfielda potrafi dokonywać asocjacji tak jak nasz mózg.
- Asocjacja oznacz, że przechowuje się obrazy (lub inne złożone informacje) może być rozpoznany nie tylko z pełnych danych, ale również z małego jego fragmentu lub zaszumionego.
- Przypomnienie następuje poprzez pamiętanie pewnych cech: np.. koloru włosów, zapachu, oczu, kształtu nosa. Cechy te są powiązane, choć przechowywane są w różnych częściach mózgu (w sieci w różnych neuronach).

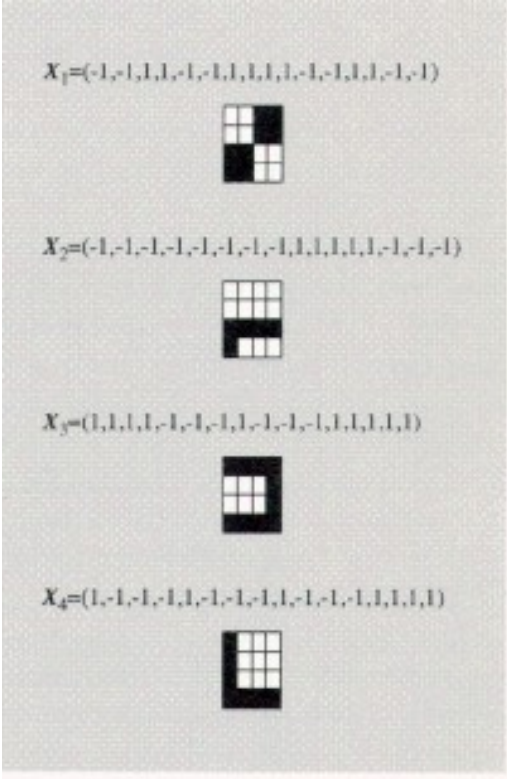
Cechy sieci Hopfielda

- Jest jednowarstwowym modelem z taką ilością neuronów jak liczba wejść (cech).
- Ponieważ każdy neuron jest połączony z każdym *m*m* wag opisuje *m* węzłów sieci.
- Oryginalny model Hopfielda pracował z danymi wejściowymi (+ 1 or - 1) (f). Wyjście liczone było według wzoru:

$$y = signum\left(\sum_{i=1}^{n} w_i u_i\right)$$

Uczenie sieci Hopfielda

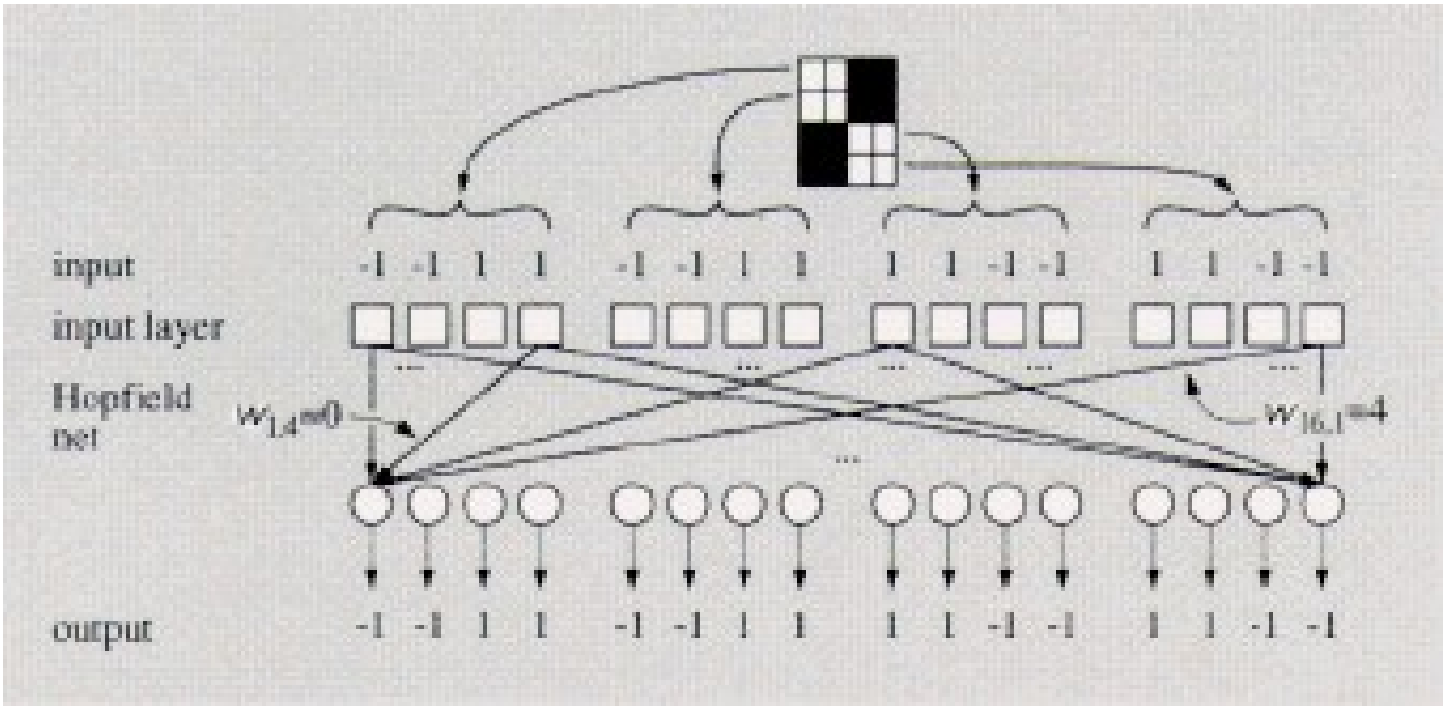
- Przykład: Sieć ma się nauczyć następujących wzorców:
- Każdy obrazek to matryca 4 x 4 (piksele) które są albo czarne (+1) albo białe (-1) = wektor 16 elementowy.



http://www2.chemie.uni-erlangen.de/publications/ANN-book/publications/ACIEE93_503/1_2_Intro_Neuron.html#TopOfPage

Architektura sieci Hopfielda dla przykładu

- Wartości wyjściowe są dokładnie równe wartościom wejściowym.



Dobór wag

- Wagi **W** w sieci Hopfielda nie muszą być dobierane w kosztownym iteracyjnym procesie uczenia. Oblicza się je bezpośrednio z prezentowanego wzorca.
- Jeżeli prezentowanych jest **p** wzorców, to wagi $\mathbf{w_{ji}}$ neuronu **j** mogą być wyprowadzone z wartości wejściowych pojedynczego wzorca **s** i stąd **Xs** według wzorów:

$$w_{ji} = \sum_{s=1}^{p} x_{sj} x_{si} \quad dla \quad j \neq i$$

$$w_{ji} = 0 \qquad dla \quad j = i$$

Interpretacja wzoru

- Wagi $\mathbf{w_{ji}}$
  - zwiększają się o 1, jeżeli w konkretnym wzorcu piksele **j** i **i** są albo jednocześnie białe, albo czarne
  - zmniejszają się o 1, jeżeli we wzorcu **i-**ty i **j-**ty piksel są różnowartościowe.
- Im więcej wzorców, w których piksele **j** i **i** pasują tym większe wagi $\mathbf{w_{ji}}$.
- W rozpatrywanym przykładzie wagi to macierz wielkości 16 x 16 (16 neuronów i każdy ma 16 wag).

Test stabilności sieci

- Wektor wejściowy (np., jeden z obrazków prezentowanych wcześniej do pamiętania) jest wprowadzany do sieci Hopfielda.
- Obliczane są wyjścia.
- Wyjścia te porównane są wejściem.
  - Jeżeli wartości są takie same, to zatrzymujemy proces.
  - W przeciwnym przypadku wartości wyjściowe przekazywane są na wejścia i proces się powtarza
  - Jeżeli po kilku cyklach otrzymamy takie same wartości na wyjściu jak podano za pierwszym razem na wejście, to sieć jest stabilna. (Oczywiście nie jest to cel sam w sobie)

Warunki stabilności

- Wyjście neuronu m nie jest kierowana na jego wejście: $w_{mm}=0$.
- Wagi połączeń neuronu m i k są symetryczne: $w_{mk}= w_{km}$

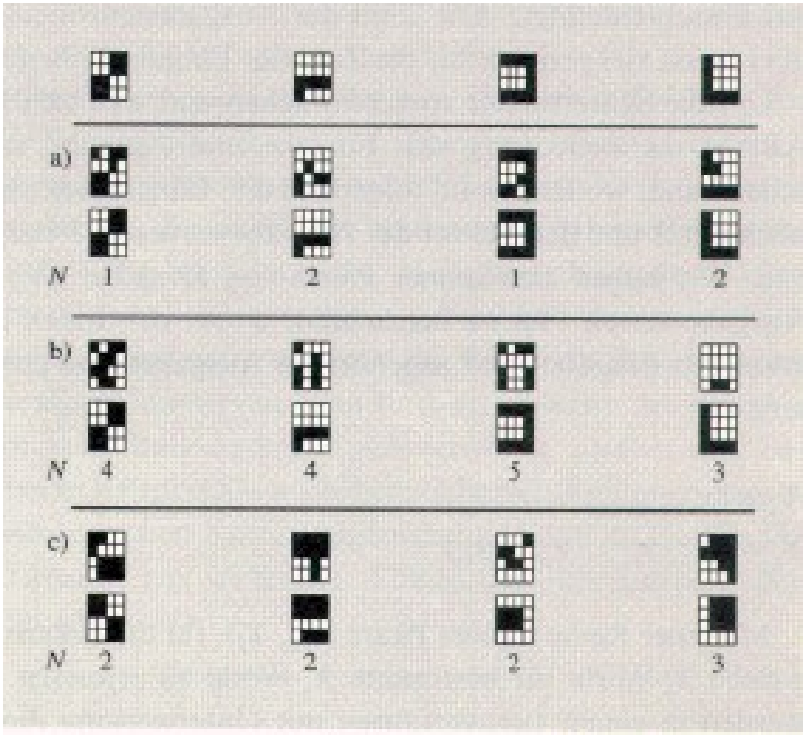
Zastosowanie

- Odzyskiwanie oryginalnych danych z niepełnych i zaszumionych danych. np.. Można uzyskać oryginalny obrazek z rozmazanego lub zepsutego.

a),b) zaszumienie <=31 %

N= iteracje

c) duże zaszumienie – błędne wzorce, lub cykle pomiędzy dwoma wzorcami
c) zaszumienie > 81%
wzorzec w negatywie



Funkcja energii

- Stan sieci Hopfielda składającej się z *N* węzłów w czasie *t* można opisać przez wektor $(y_1(t),\ldots, y_N(t))$, gdzie $y_i(t)$ jest wyjściem i-tego neuronu w czasie t.
- Energię sieci w jednostce czasu *t* definiuje się jako:

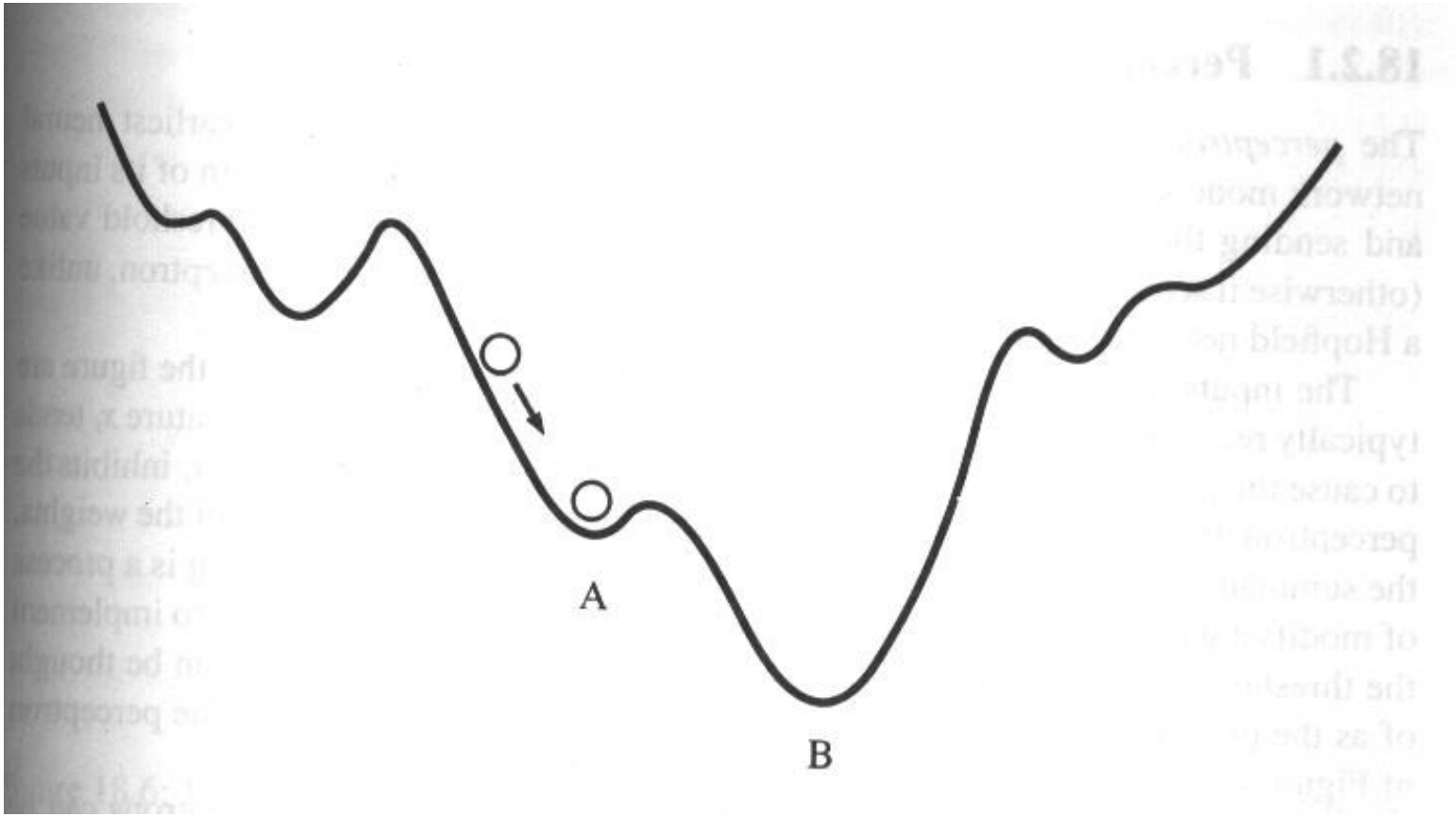
$$E(t) = -\frac{1}{2}\sum_i \sum_{j\neq i} w_{ij}\, y_i(t)\, y_j(t) + \sum_i y_i(t)\,\theta_i$$

gdzie θ_i jest progiem *i*-tego neuronu.

Funkcja energii

- Działanie sieci Hopfielda może być utożsamiane z procesem minimalizacji funkcji energii.
- Sieć sprowadza poziom energii do najbliższego wzorca przykładowego.
- Pozycje minimów funkcji energii są zdeterminowane przez wartości wag.
- Sieć może zapamiętać liczbę wzorców, która jest równa w przybliżeniu 15% liczby węzłów – duża wada. Przy dużej liczbie pamiętanych wzorców następuje gwałtowny wzrost rozmiarów macierzy wag.

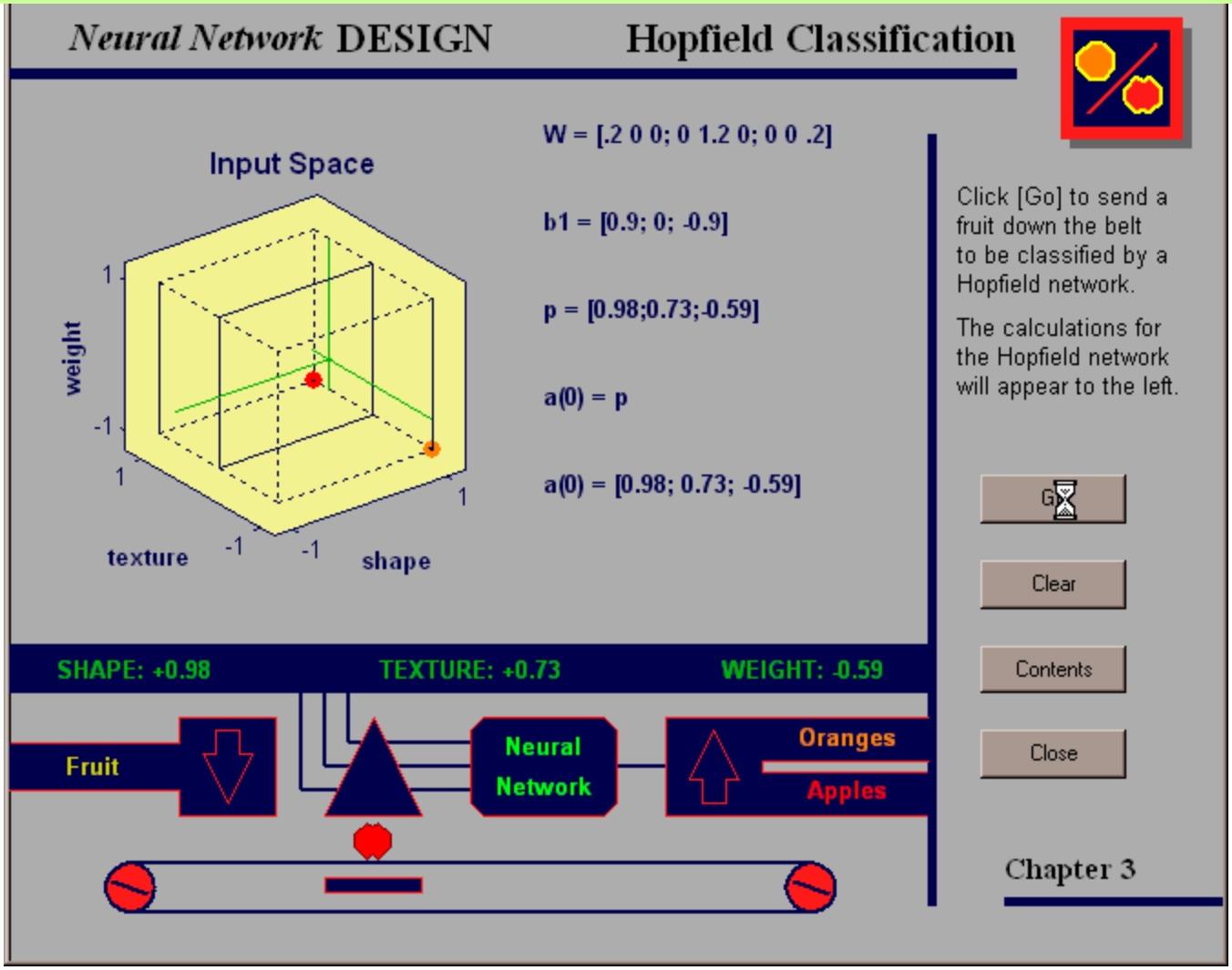
Proces relaksacji



Przykład klasyfikacji

- Rozpoznajemy jabłka i pomarańcze na podstawie 3 cech: waga, faktura i kształt.
- Zastosowano w sieci 3 neurony.
- W pierwszym kroku jako wyjście podane są cechy wejściowe (odczytane z czujnika).
- Udzielana przez sieć odpowiedź to współrzędne punktu w przestrzeni wejść oznaczone jako idealna pomarańcza i idealne jabłko.

Pierwszy krok rozpoznania



Zastosowanie sieci Hopfielda

- Rozpoznawanie wzorców (pisma,obrazów, mowy).
- Analiza inwestycji (wahania kursów akcji, klasyfikacja obligacji, ocena stopnia opłacalności przedsięwzięć inwestycyjnych, wybór funduszu powierniczego, wybór strategii sprzedaży i inne).
- Kontrola procesów.
- Optymalizacja – problemy NP-zupełne.