

prof. dr hab. inż. **Andrzej Kos**

Tel. 34.35, email: kos@agh.edu.pl

Pawilon C3, pokój 505

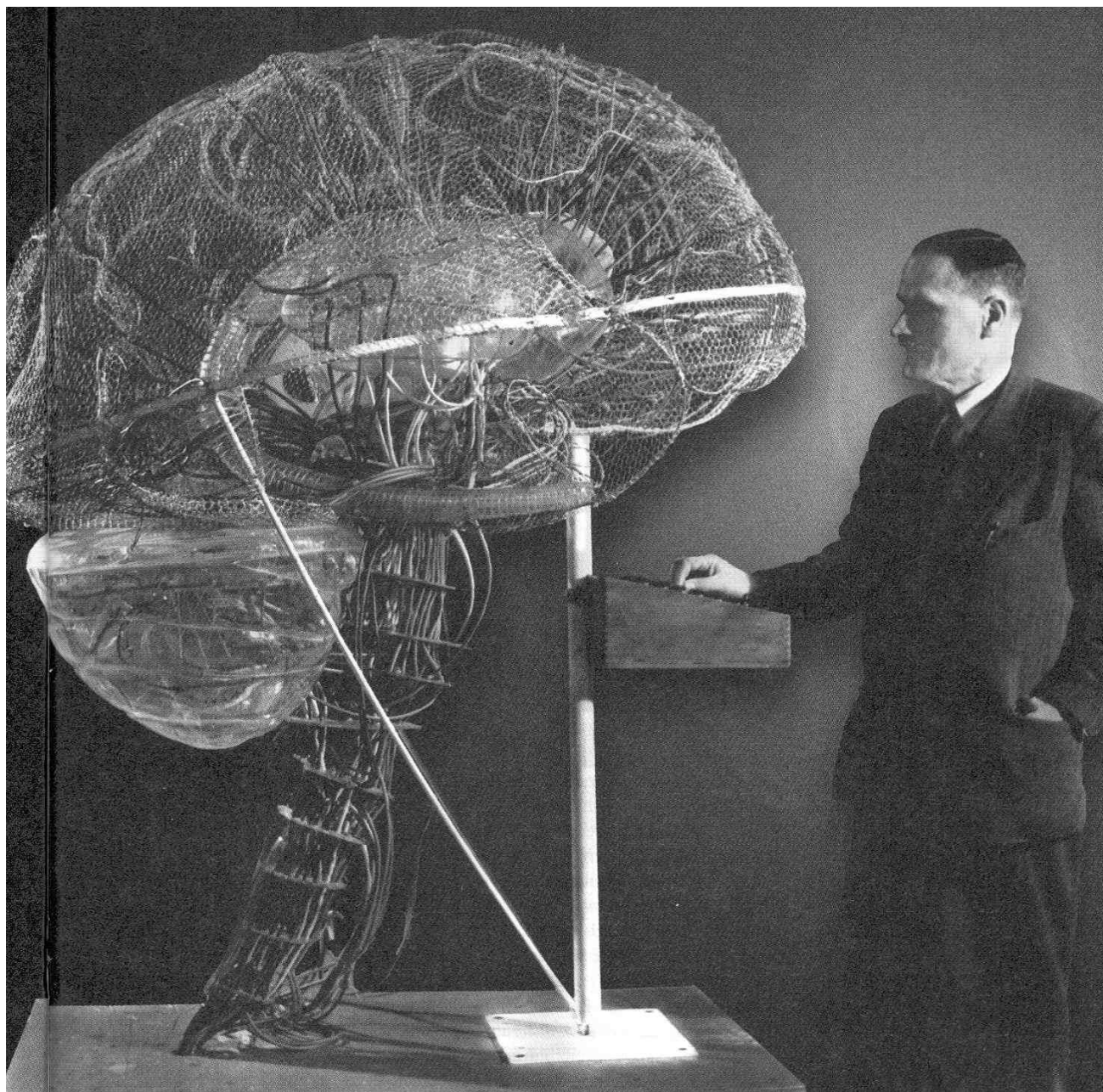
PRZEMYSŁOWE ZASTOSOWANIA SZTUCZNEJ INTELIGENCJI

Forma zaliczenia: egzamin

UWAGA !

Zamieszczony dalej konspekt wykładu stanowi skrót treści przewidzianej programem nauczania

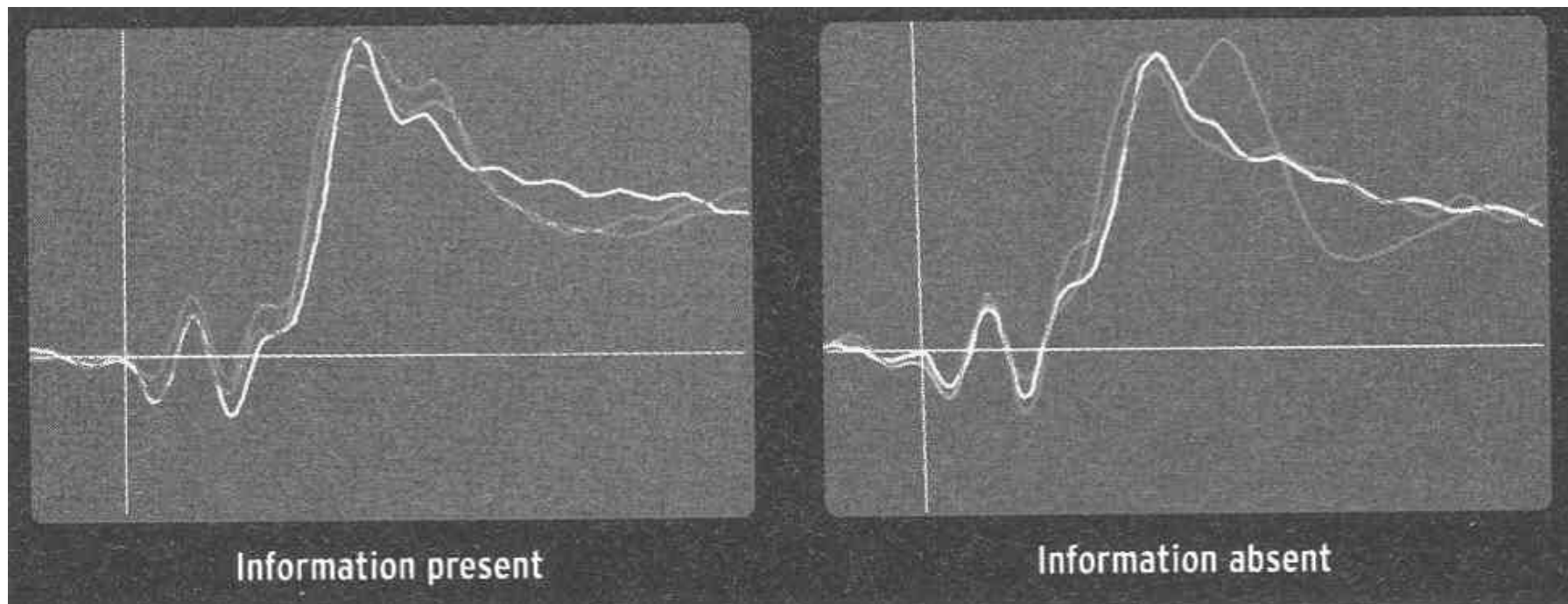
Pełna wiedza przedstawiona zostanie podczas kolejnych wykładów



Niedościgniony wzorzec – generator pomysłów



Sygnały elektryczne mózgu a prace inżynierskie



Wykorzystanie indywidualnych cech sygnałów mózgu do identyfikacji personaliów

**10 Obszarów
mózgu, w których
można wykryć
różne
schorzenia**

**Analiza
sygnałów
bio-elektrycznych**

**Wnioski
medyczne**

**Wnioski
inżynierskie**

- ▲ Increased activity
- ▼ Decreased activity
- Equivalent

PFC = Prefrontal cortex

PTSD = Post-traumatic stress disorder

Brain Region	Autism	Schizophrenia	Depression	PTSD
Orbitofrontal cortex				▲
Amygdala	▼	▼	▲	▲
Fusiform gyrus	▼			
Ventrolateral PFC	▼	▼		
Subgenual cingulate			▲	▼
Rostral cingulate	▼		▲	▼
Superior temporal sulcus	▲ or ■			
Dorsomedial PFC	▼	▲		▼
Dorsolateral PFC	▼	▲ or ▼	▼	
Dorsal cingulate	▼	▼	▲ or ▼	▲ or ▼

Source: Dr. Cheryl Grady, Rotman Research Institute

Obszar przedśrodkowy
sterujący ruchami ciała

Obszar zaśrodkowy

czucie dotyku

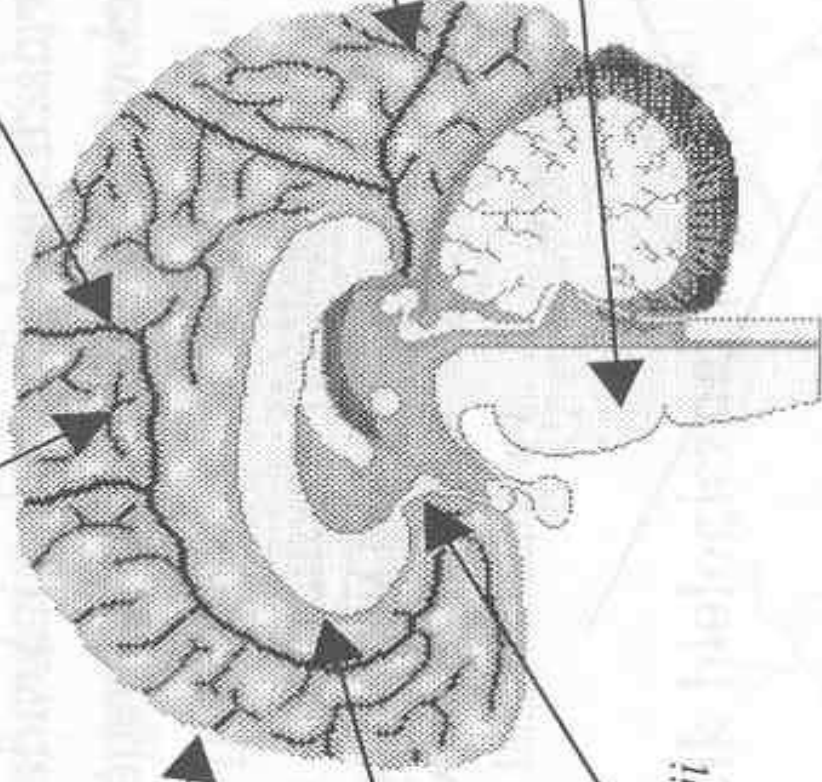
Płaty czołowe
siedlisko inteligencji?

Spoidło wielkie
łączy półkule

Zwoje potyliczne
widzenie

Podwzgórze
siedlisko uczuć i emocji

Pień mózgu
regulacja funkcji
całego ciała



Definicja sztucznej inteligencji

System inteligentny = system uczący się.

Geneza NN

- Ograniczenia sprzętowe w postępie budowy tradycyjnego sprzętu komputerowego
- Mała wydajność systemów tradycyjnych
- Przewaga mózgu ludzkiego nad maszyną

Budowa mózgu a proces uczenia się

- Dwie półkule mózgowe jako „procesory”
 - lewa - algorytmy
 - prawa - koncepcje, obrazy
- Rytm pracy mózgu:
 - Szybki – algorytmy
 - Wolny – praca twórcy (rola muzyki)
 - Bardzo wolny (wizje)
- Współpraca obu półkul mózgu

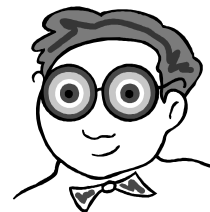
Dane mózgu

- Liczba komórek neuronowych mózgu człowieka, ok. 10^{10}
- Liczba połączeń między komórkami, ok. 10^{18}
- Przeciętna odległość między komórkami 0,01-1 mm
- Impulsy elektr. Generowane przez komórki: $f=1-100\text{Hz}$, $U=0-100\text{mV}$, czas trwania=1-2ms
- Szybkość propagacji sygnałów 1-100m/s
- Szybkość pracy mózgu: $10 \text{ operacji} \times 10^{15} \text{ synaps} \times 100\text{Hz} = 10^{18} \text{ operacji/s}$

Mózg v. superkomputer

$$\frac{\text{Mózg}}{\text{ETA10}} = \frac{10^{18} \text{ operacji/s}}{10^{10} \text{ operacji/s}} = 10^8$$

czyli



= 100 000 000



Pionierskie prace nad sztuczną inteligencją:

-Początki lata 30 minionego stulecia.

-McCulloch i Pitts przedstawili sformalizowany model neuronu w roku 1943

Fundamentalnym elementem modelu jest sumowanie sygnałów wejściowych z odpowiednią wagą i poddanie otrzymanej sumy działaniu liniowej lub nieliniowej funkcji aktywacji

$$y_i = f \left[\sum_{j=1}^N w_{ij} x_j \right]$$

-Kilka lat później Hebb przedstawił własną teorię uczenia – doboru wag W_{ij} . Wykorzystał to do budowy pamięci asocjacyjnych. Przyrost wagi w procesie uczenia jest proporcjonalny do iloczynu sygnałów wyjściowych neuronów połączonych wagą W_{ij}

$$W_{ij}(k+1) = W_{ij}(k) + \eta y_i(k) y_j(k)$$

-1957, Rosenblatt skonstruował fizyczny model neuronu jako system elektromechaniczny (potencjometry, dźwignie, silniki elektryczne itp.) do rozpoznawania znaków alfa-numerycznych

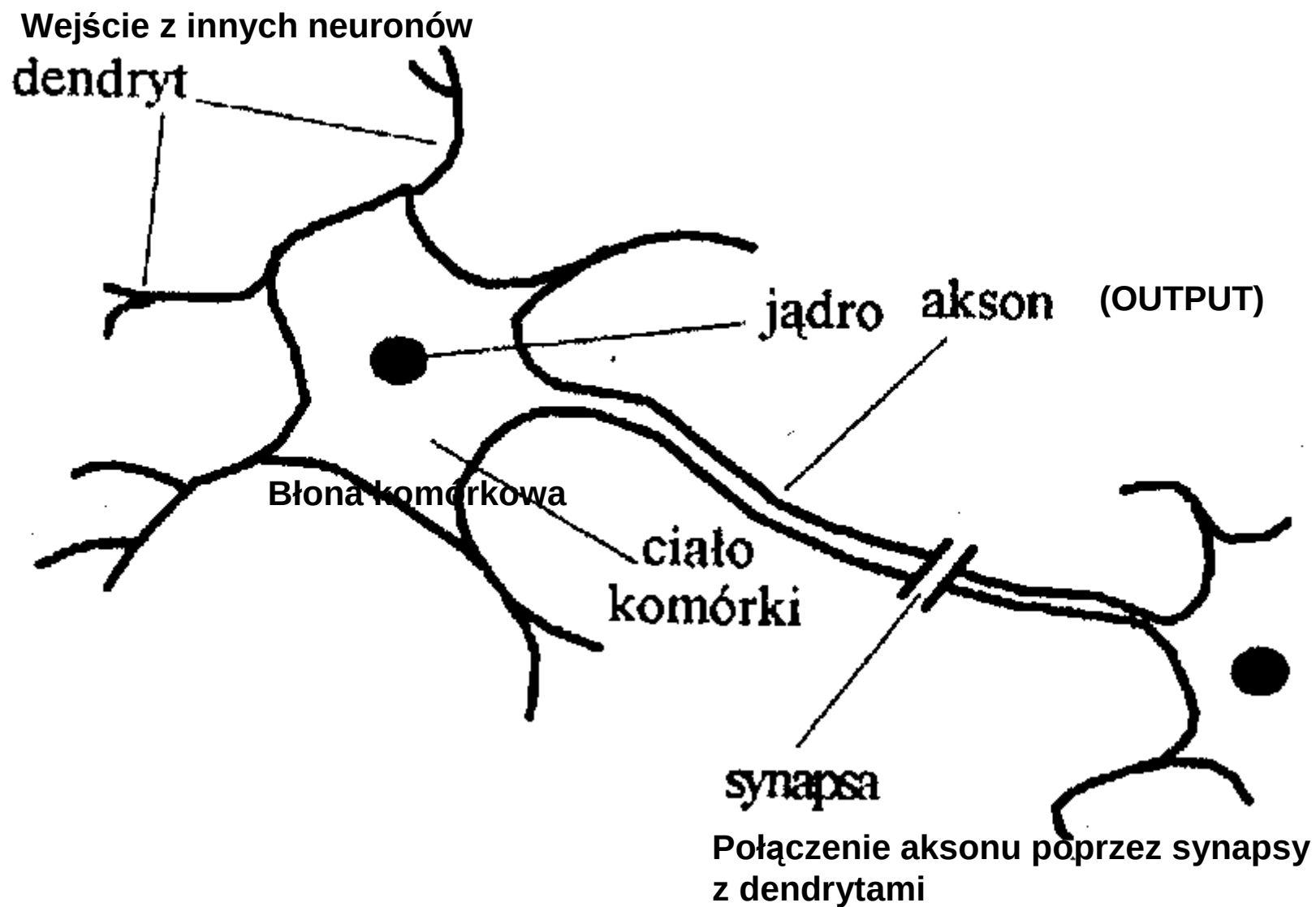
-W Polsce pierwsze prace nad nn: Karpiński i Michalski, z Instytutu Automatyki PAN

Funkcje sieci neuronowych:

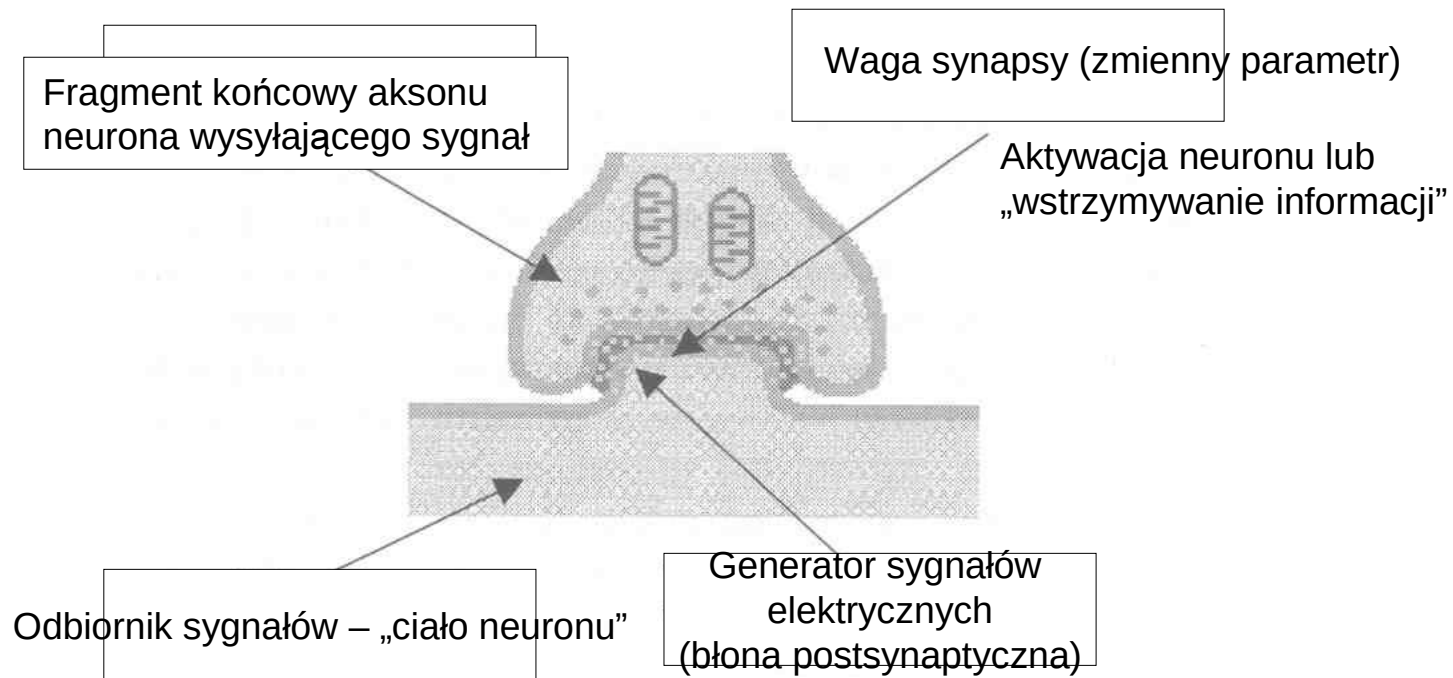
- Aproksymacja
- Klasyfikacja i rozpoznawanie wzorców
- Predykcja
- Asocjacja
- Sterowanie

Budowa pamięci niekonwencjonalnych

- Pamięć klasyczna (np.: drzewo, algorytm). Informacja zawarta jest tylko w określonej części pamięci
- Pamięć asocjacyjna – znakomita większość pamięci zaangażowana jest do zapamiętania każdej informacji (mechanizm równoległy)



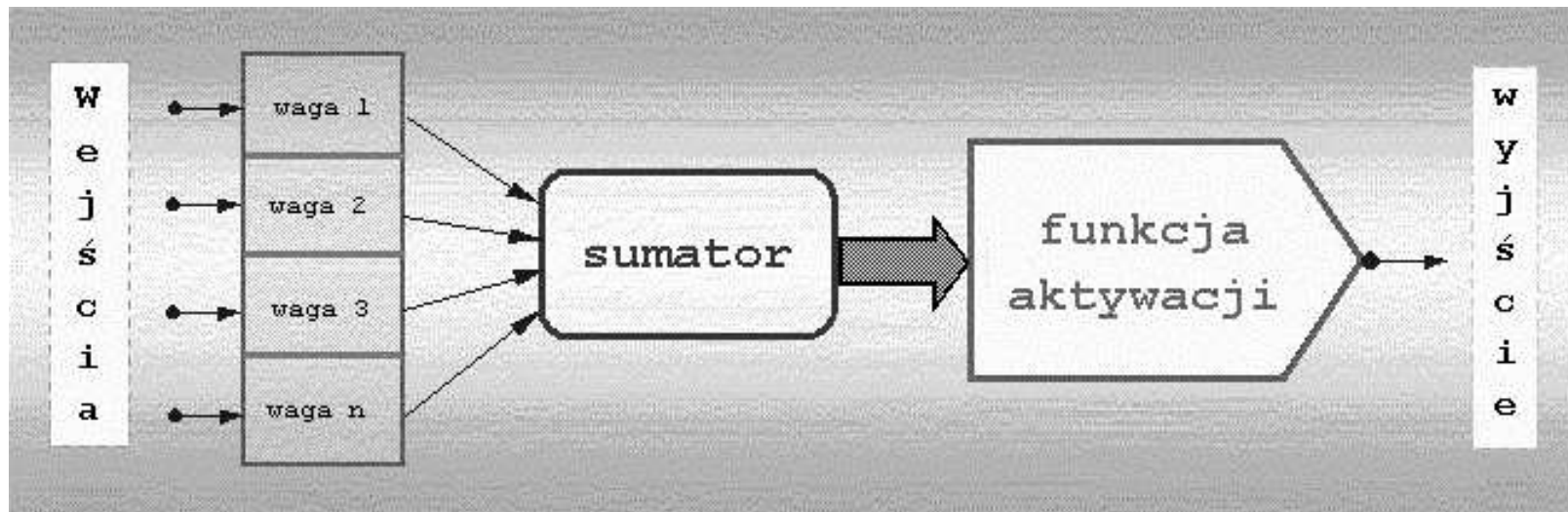
Uproszczona budowa komórki nerwowej - neuronu



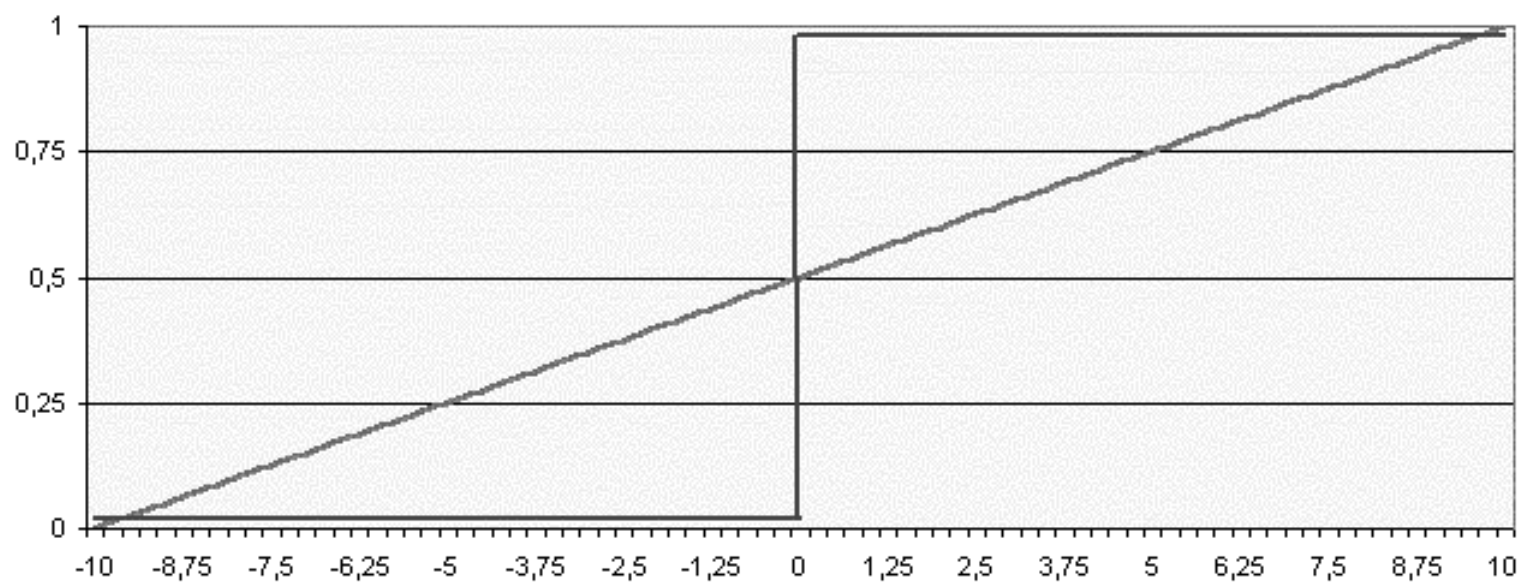
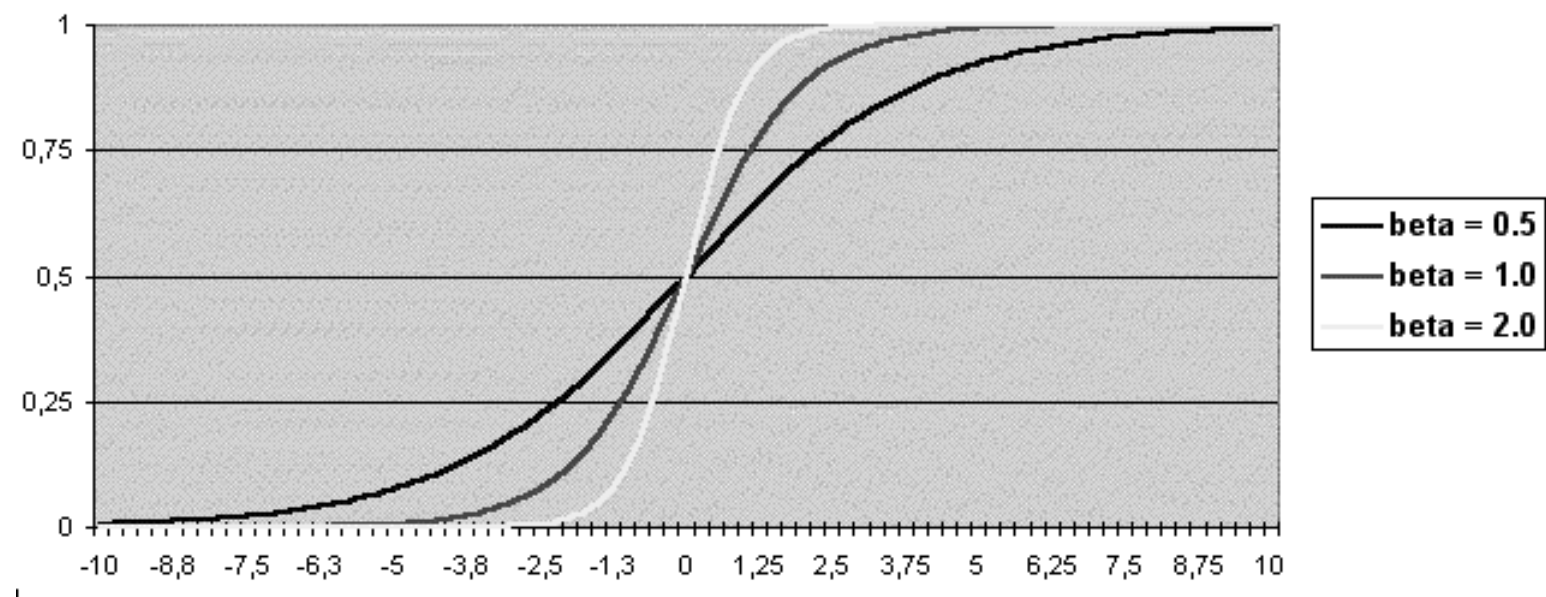
Budowa synapsy komórki nerwowej (biologicznej)

Cechy neuronu:

- Ciało komórki gromadzi sygnały pochodzące od innych neuronów poprzez dendryty i dokonuje ich sumowania (całkowania)
- Ważną rolę odgrywa opóźnienie czasów sygnałów
- Jeśli całkowity (sumaryczny) sygnał wejściowy jest mniejszy od wartości progowej to neuron pozostaje nieaktywny
- Przekroczenie wartości progowej generuje sekwencję sygnałów o okresie ok. 1 ms
- Każdy neuron mózgu ludzkiego otrzymuje sygnały wejściowe od 10^3 do 10^4 innych neuronów
- System neuronowy działa asynchronicznie (bez zegara)
- Specyficzna cechą nn jest zdolność adaptacji i uczenia się
- Uczenie, w uproszczeniu, polega na zmianie połączeń synapsowych zgodnie z regułą Hebba
- Nauka jest procesem lokalnym i wymaga powtórzeń



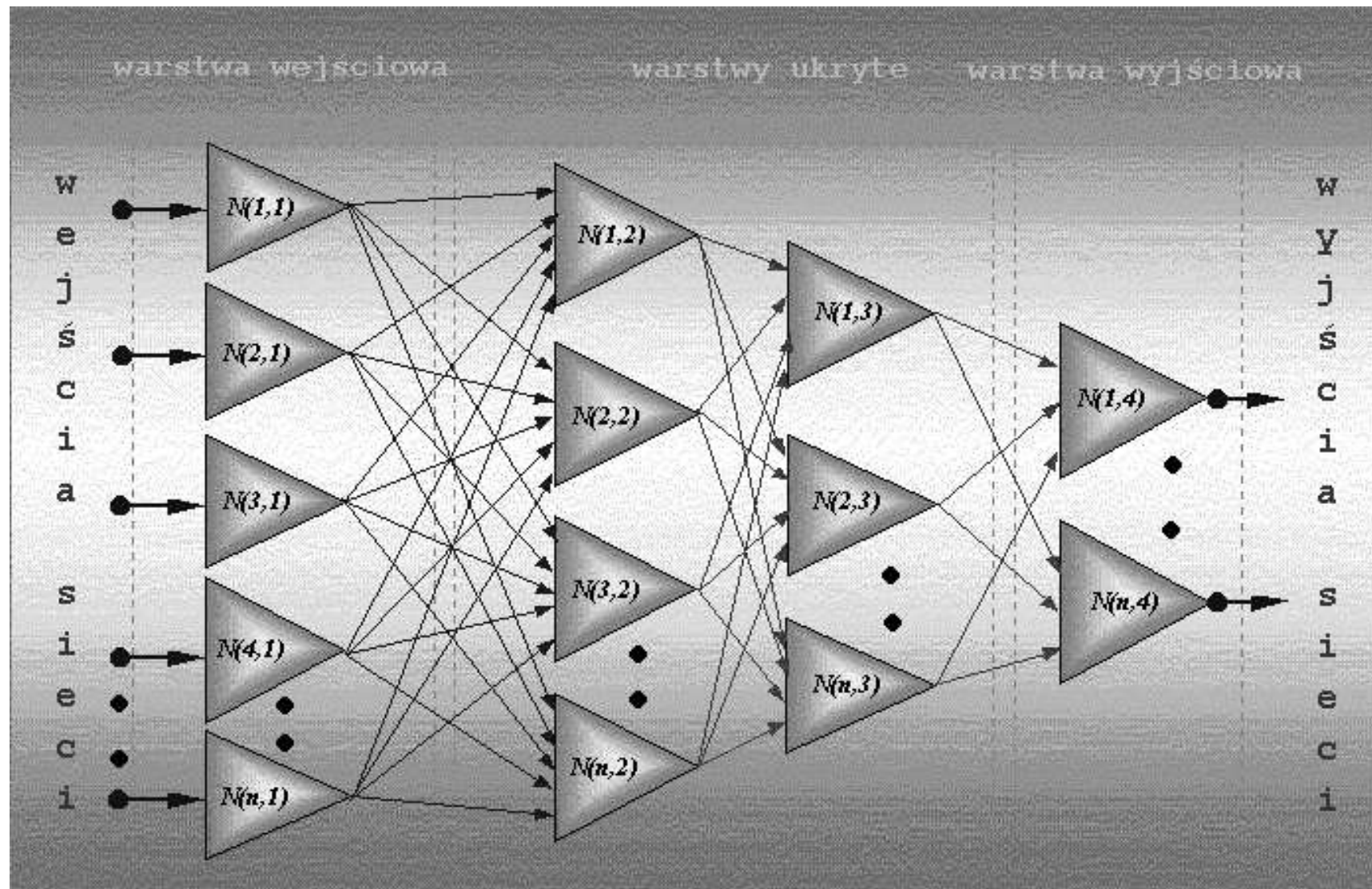
Model neuronu



Funkcje aktywacji neuronu

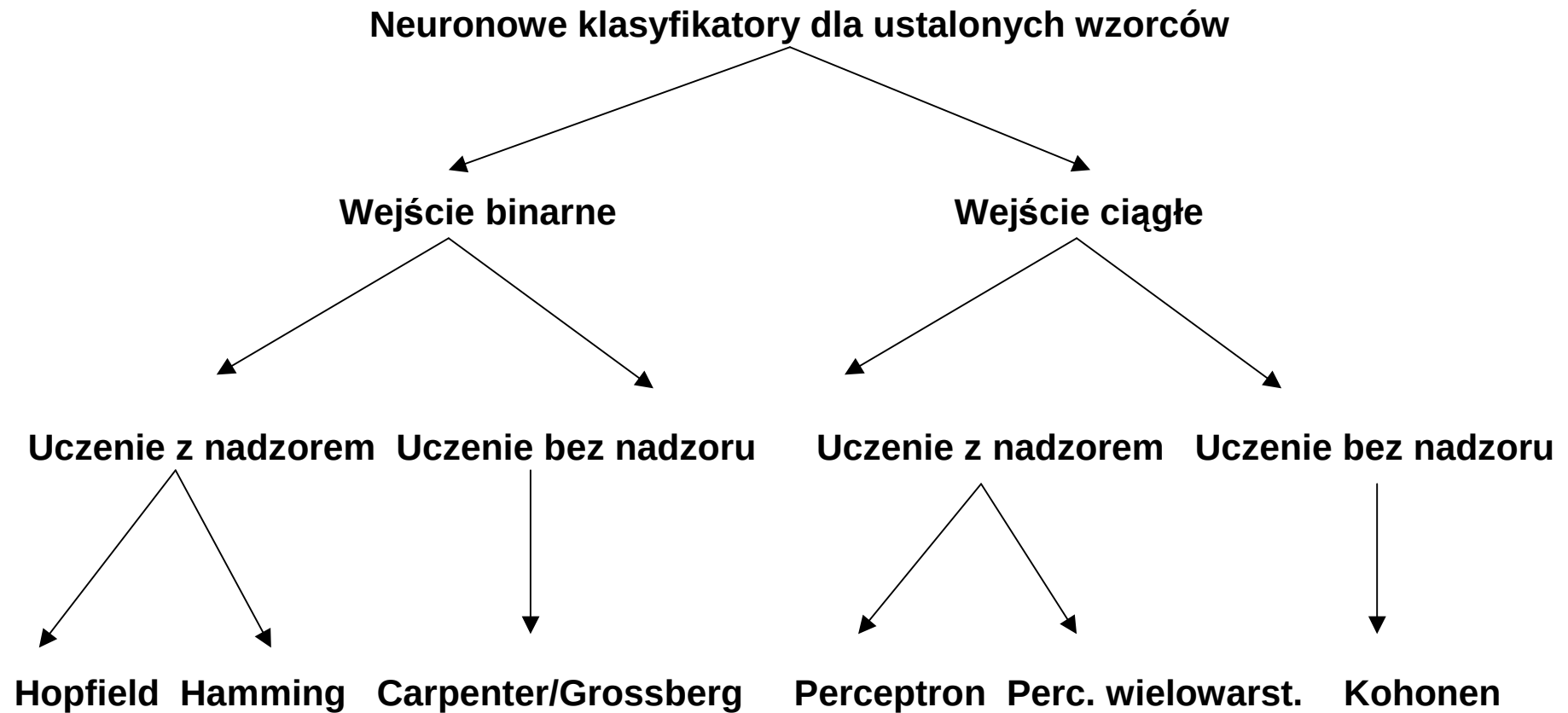
funkcja	Wzór funkcji	Wzór pochodnej
Sigmoida	$f(x) = \frac{1}{1 + e^{-(beta * x)}}$	$f'(x) = beta * (1 - f(x)) * f(x)$
Tangens hiperboliczny	$f(x) = \tanh(beta * x)$	$f'(x) = beta(1 - f^2(x))$
Sinusoida	$f(x) = \sin(beta * x)$	$f'(x) = beta\sqrt{1 - f^2(x)}$
Cosinusoida	$f(x) = \cos(beta * x)$	$f'(x) = -beta\sqrt{1 - f^2(x)}$
$\frac{x}{(1 + x)}$ (bez nazwy)	$f(x) = \frac{beta * x}{(1 + beta * x)}$	$f'(x) = \frac{beta}{1 + beta * x } - \frac{ beta * x }{(1 + beta * x)^2}$

Zapis różnych funkcji aktywacji

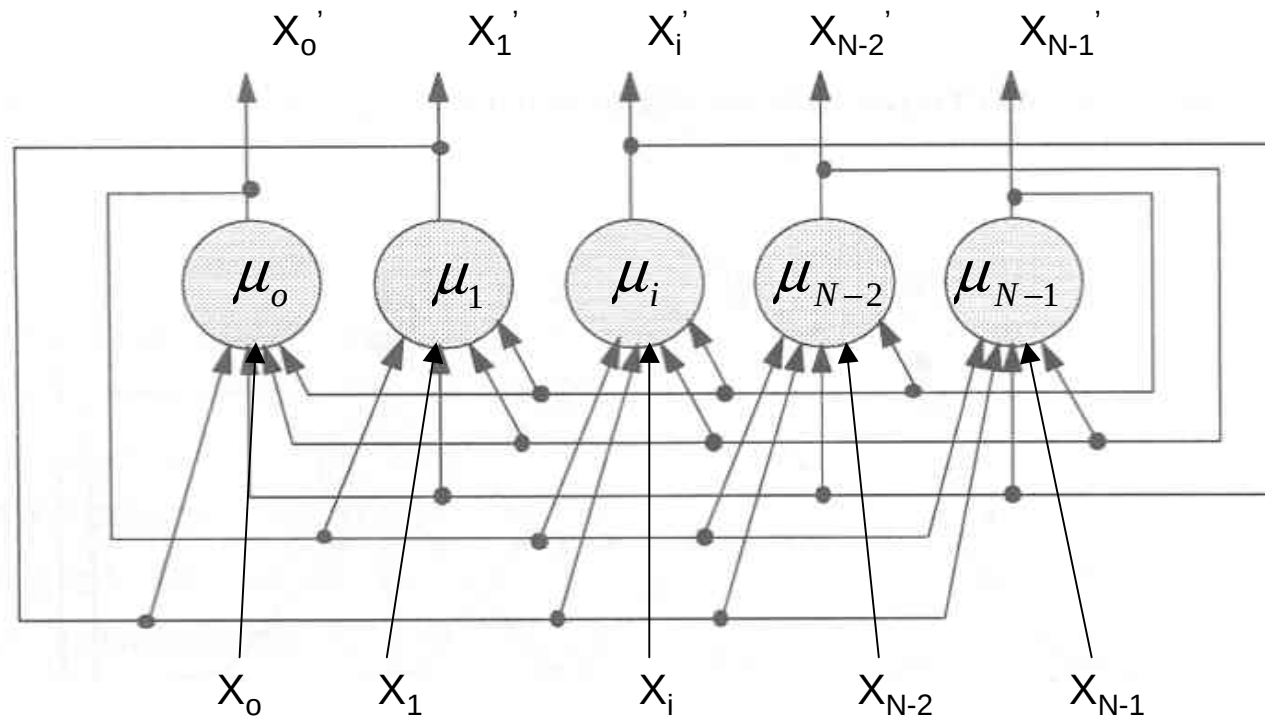


Perceptron wielowarstwowy

NEURONOWE KLASYFIKATORY



Sieć Hopfielda



Zastosowanie: optymalizacja, pamięć asocjacyjna

Np. wejście binarne, 8-bitowa reprezentacja znaków ASCII

Charakterystyka neuronu typu „hard limiting”

Algorytm funkcjonowania sieci Hopfielda jako klasyfikatora

Krok 1: Zdefiniowanie wag połączeń (kodowanie wzorców)

$$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & i \neq j \\ 0, & i = j, \quad 0 \leq i, j \leq N-1 \end{cases}$$

t_{ij} – Waga połączenia węzła i z węzłem j .

x_i^s – wartość (-1 lub 1) elementu i -tego należącego do klasy s

Krok 2: Start algorytmu

Wprowadzenie wartości wejściowych rozpoznawanej figury

$$\mu_i = x_i, \quad 0 \leq i \leq N-1$$

Gdzie $\mu_i(t)$ jest wyjściem neuronu i w czasie t

x_i – elementem i wejściowej figury, $x_i=1$ lub -1 .

Krok 3: Wykonaj kolejne iteracje

$$\mu_j(t+1) = f_h \left[\sum_{j=0}^{N-1} t_{ij} \mu(t) \right], \quad 0 \leq j \leq N-1$$

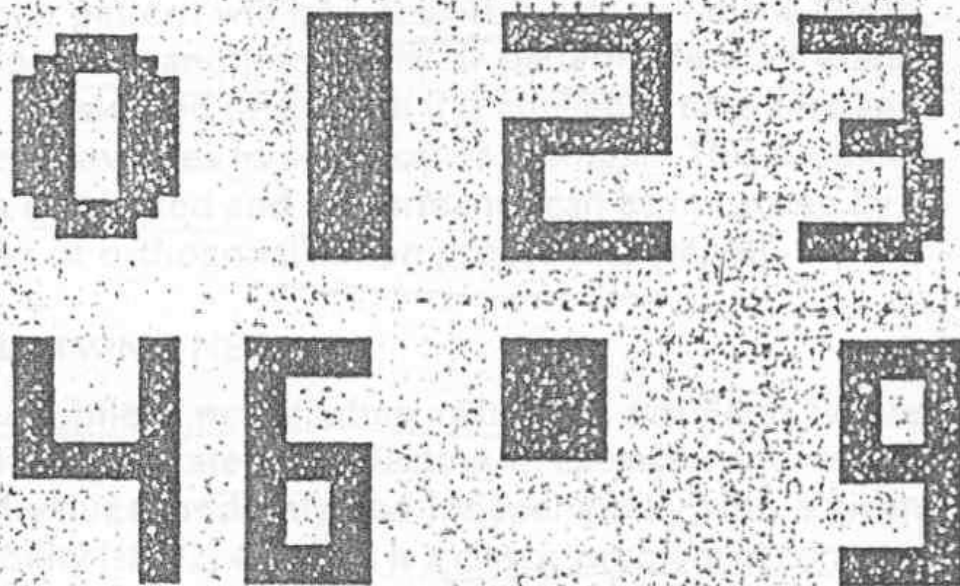
funkcja f_h jest tzw. „hard limiting” funkcją nieliniową
Proces jest powtarzany dopóki wyjścia węzłów nie przyjmują ustalonych wartości. Wówczas wyjście węzła reprezentuje wzorec najbardziej pasujący do nieznanej figury wejściowej.

Krok 4: Idź do punktu 2

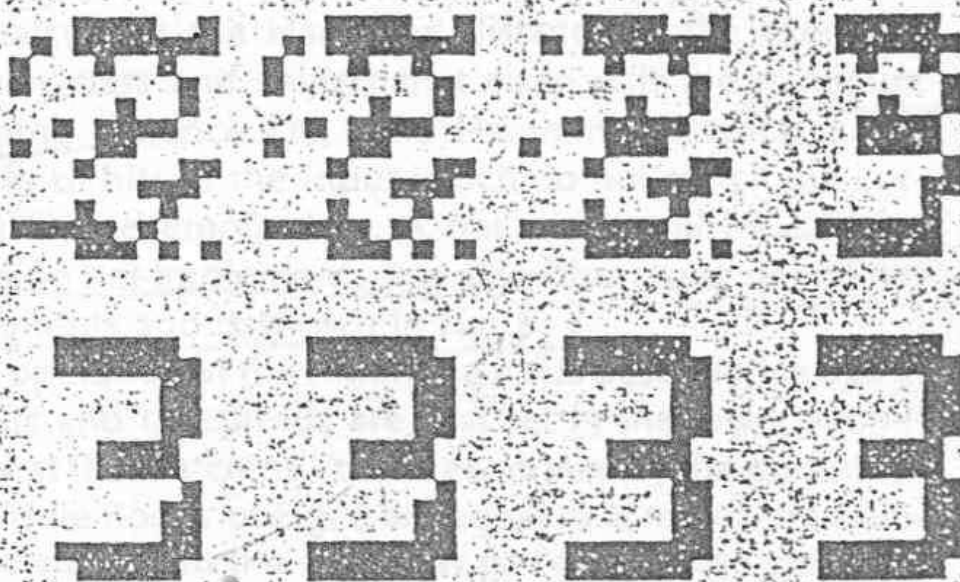
Uwaga: -liczba wzorców jest ograniczona. Jeśli zbyt duża sieć zbiega się do nowego nieoczekiwanego wzorca.

- Optymalna liczba wzorców = $0.15 N$ (liczby neuronów). Np. 10 klas wymaga $N=70$ i ponad 5000 wag.
- Problemy pojawiają się z rozpoznaniem właściwego wzorca, gdy wzorce są podobne.

A) EIGHT EXEMPLAR PATTERNS



B) OUTPUT PATTERNS FOR NOISY "3" INPUT

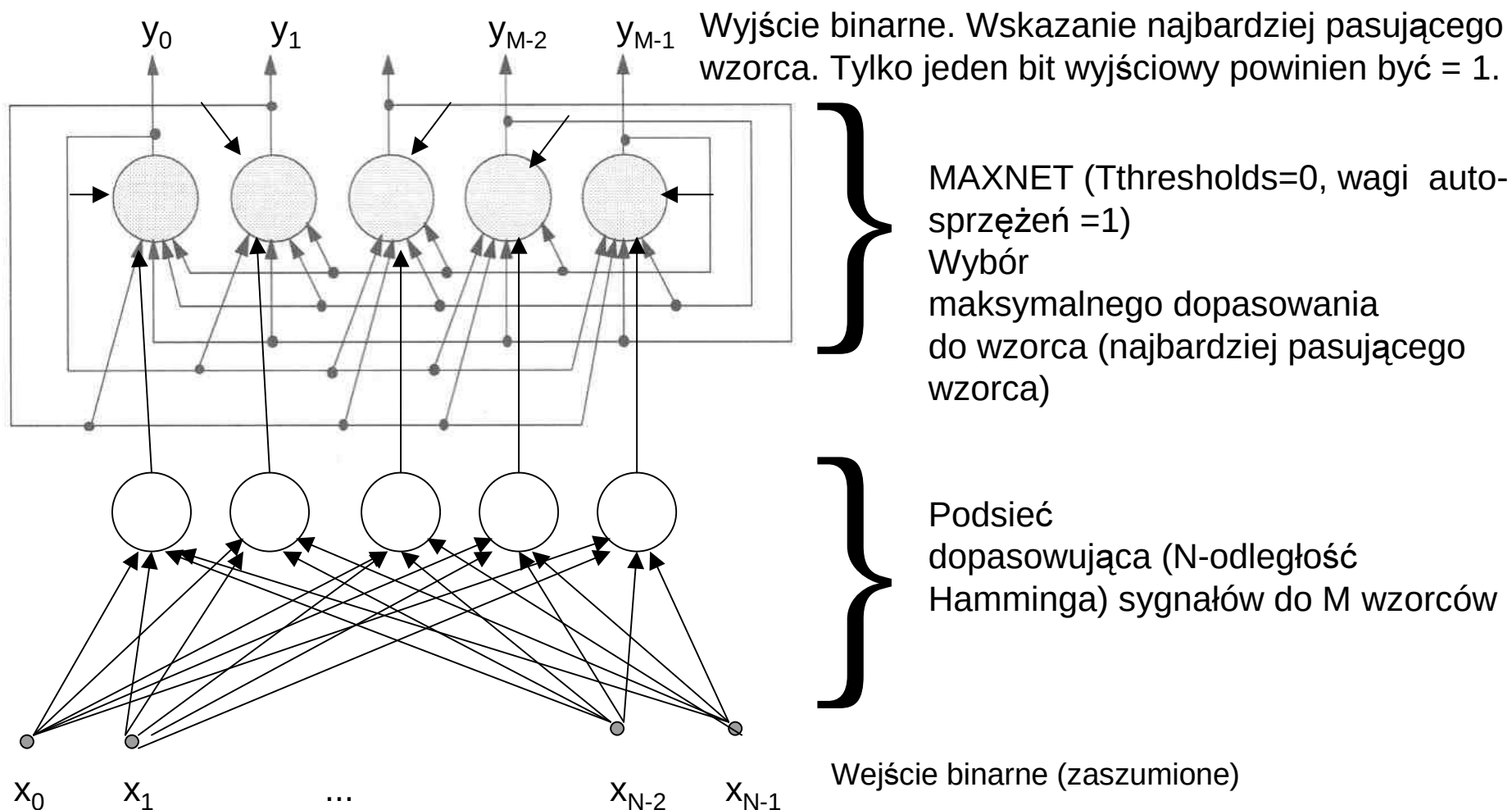


Przykład rozpoznawania figur przy pomocy sieci Hopfielda

Sieć składała się z 120 neuronów,
liczba wzorców = 8 - rys A.
Wyniki rozpoznania na rys. B

SIEĆ NEURONOWA HAMMINGA (KLASYFIKATOR)

Def. Odległość Hamminga = liczba bitów niezgodności rozpoznawanej figury z wzorcem



Wagi sprzężeń pomiędzy neuronami przyjmują wartości $-\varepsilon$, gdzie $\varepsilon < 1/M$.

Zastosowanie np. w telekomunikacji - eliminacja zniekształceń sygnałów przez szumy

Algorytm funkcjonowania sieci Hamminga jako klasyfikatora

Krok 1: Wyznaczanie wag połączeń i rekompensat

Dla dolnego podobwodu:

$$w_{ij} = \frac{x_i^j}{2}, \quad \Theta_j = \frac{-N}{2}$$

$$0 \leq i \leq N-1, \quad 0 \leq j \leq M-1$$

gdzie w_{ij} jest wagą połączenia neuronu i z neuronem j ,
 Θ_j jest wartością progową funkcji pobudzenia neuronu.
 x_i^j jest elementem i wzorca j .

Dla górnego podobwodu:

$$t_{kl} = \begin{cases} 1, & k = l \\ -\varepsilon, & k \neq l, \quad \varepsilon = \frac{1}{M} \end{cases} \quad 0 \leq k, l \leq M-1$$

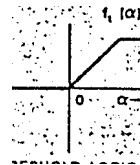
t_{kl} - waga połączeń neuronu k z neuronem l . Wszystkie wartości progowe = 0

Krok 2: Start algorytmu

$$\mu_j(0) = f_t \left(\sum_{i=0}^{N-1} w_{ij} x_i - \Theta_j \right)$$
$$0 \leq j \leq M - 1$$

$\mu_j(t)$ jest wyjściem neuronu j w górnej sieci w czasie t
 x_i jest elementem i wejścia

f_t jest funkcją neuronu:



Krok 3: Wykonaj kolejne iteracje

$$\mu_j(t+1) = f_t \left(\mu_j(t) - \varepsilon \sum_{k \neq j} \mu_k(t) \right)$$
$$0 \leq j, k \leq M - 1$$

Powtarzanie kolejnych kroków iteracji dopóki tylko jeden neuron wyjściowy przyjmuje wartość pozytywną (numer wzorca)

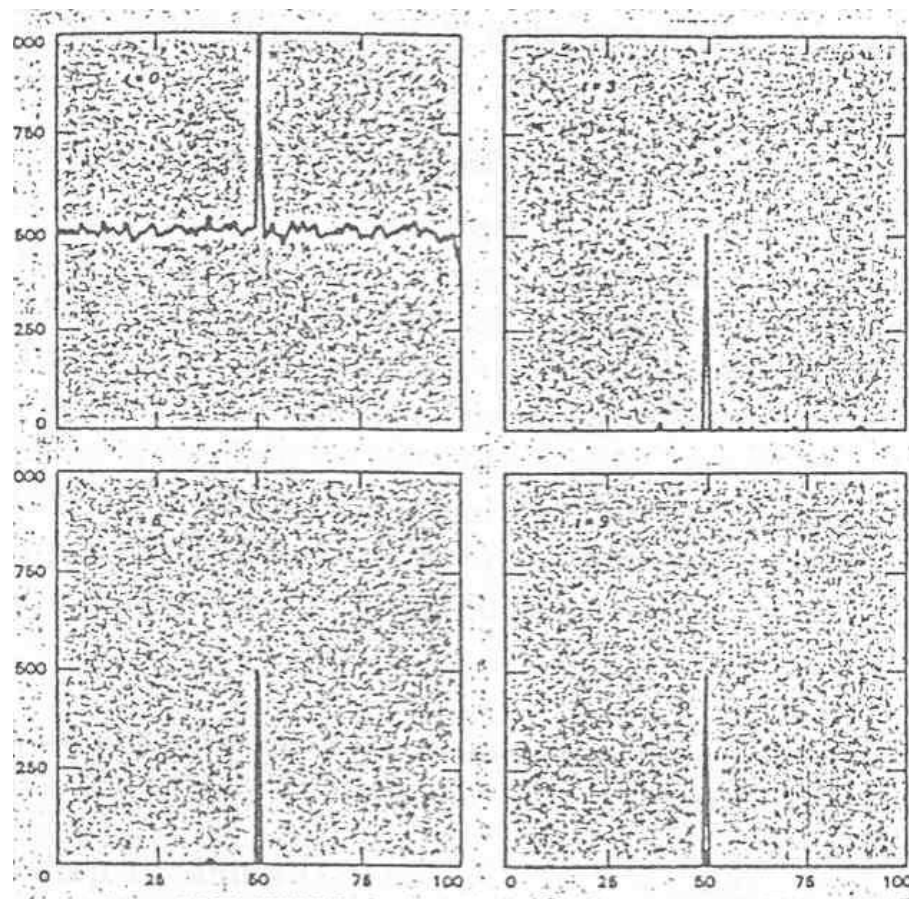
Krok 4: Idź do kroku 2

Przykład funkcjonowania klasyfikatora Hamminga

1000 bitowe wejście
100 wzorców (wyjście)

Stan wyjść dla $t=0$

po 3 iteracjach



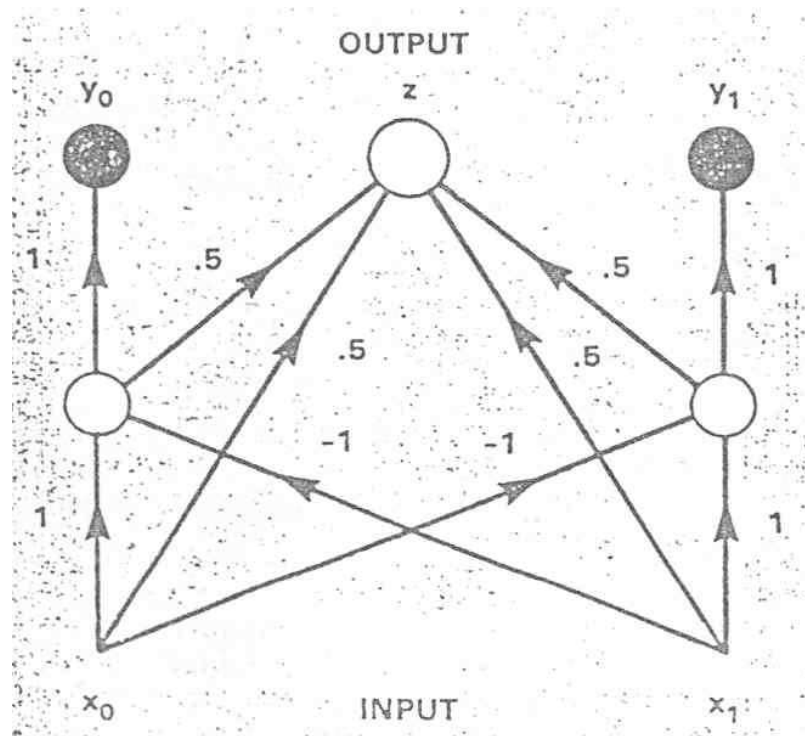
Poddany klasyfikacji przykład
pasuje najbardziej
do wzorca nr 50

Po 6 iteracjach

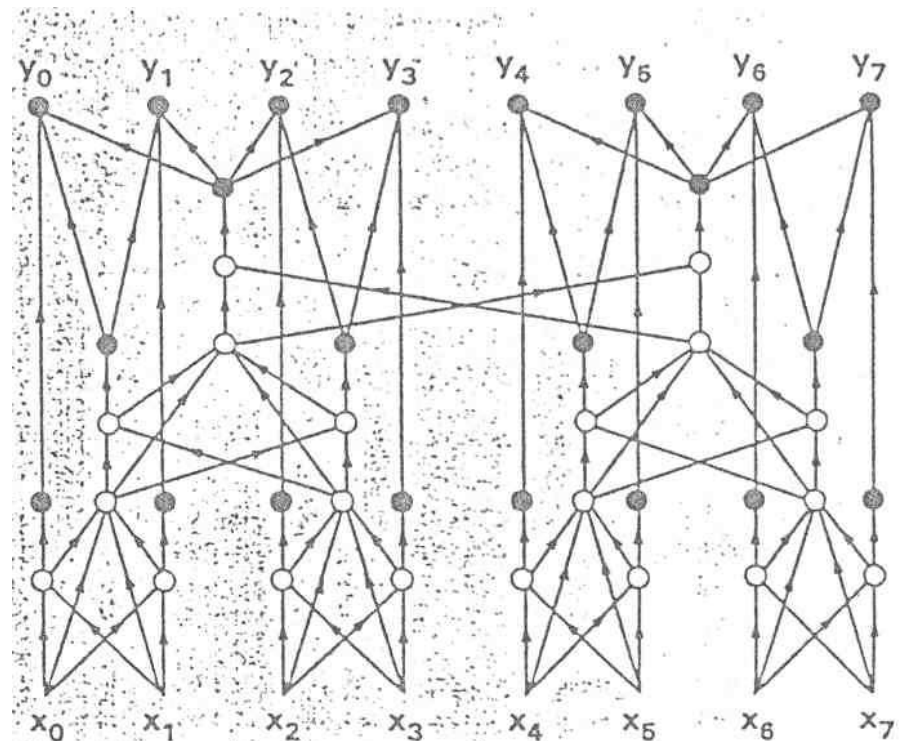
Po 9 iteracjach

Wyostrzenie odpowiedzi sygnałów wyjściowych klasyfikatorów

Problem jednoznacznego określenia wzorca, czyli pojawienia się wyraźnych 0 lub 1 w MAXNET



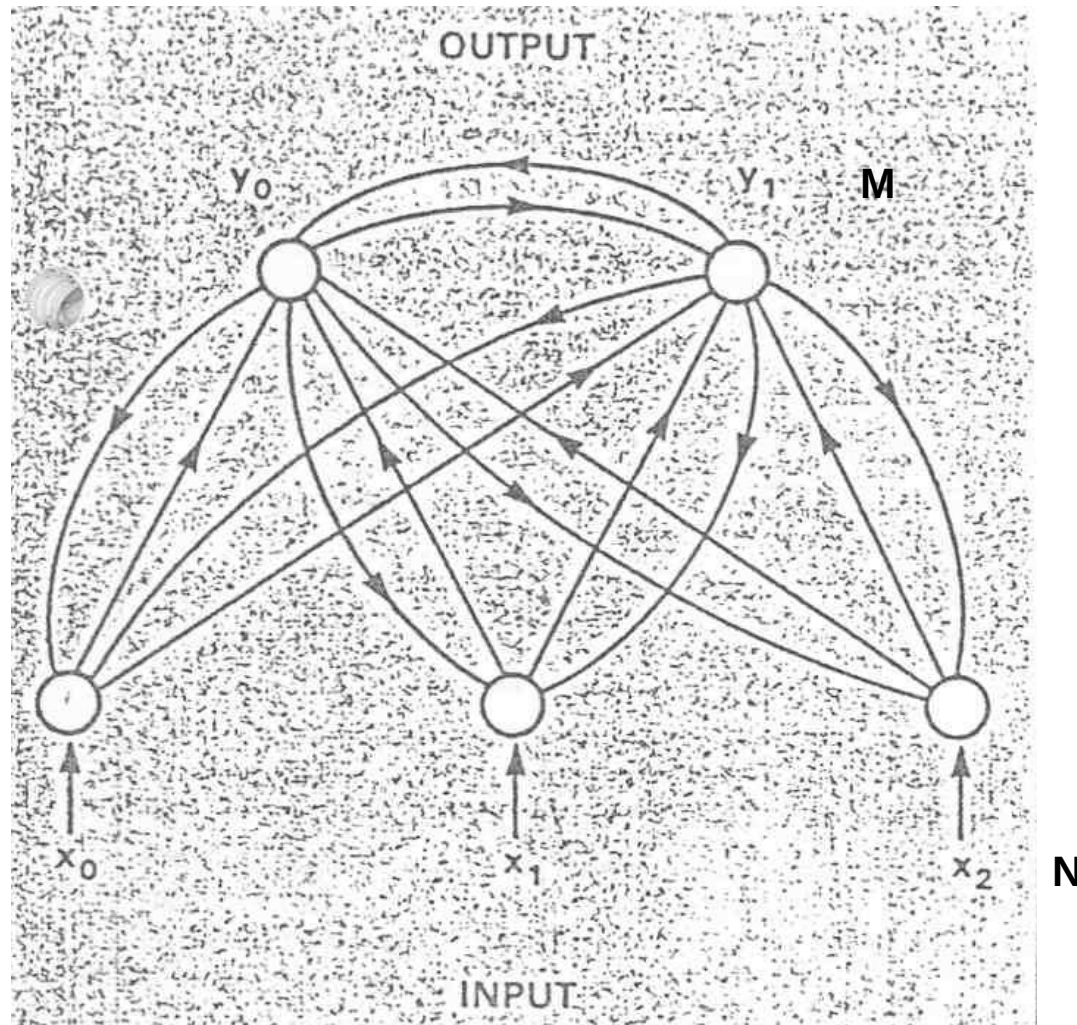
Komparator binarny wybierający maksimum z dwóch sygnałów



Komparator wybierający największy sygnał z ośmiu

KLASYFIKATOR CARPENTERA / GROSSBERGA

Sieć uczona bez nadzoru.



Wejście binarne. Na wyjściu pojawia się tylko jedna „1” określająca numer wzorca

Algorytm funkcjonowania sieci CARPENTERa/GROSSBERGa

Krok 1: Wyznaczenie wartości wag połączeń

$$t_{ij}(0) = 1$$

$$b_{ij}(0) = \frac{1}{1 + N}$$

$$0 \leq i \leq N - 1, \quad 0 \leq j \leq M - 1 \quad N\text{-we}, \quad M\text{-wy}$$

Przyjmujemy wartość ε , $0 \leq \varepsilon \leq 1$

$t_{ij}(t)$ = waga połączenia z góry do dołu pomiędzy węzłem wejściowym i , a węzłem wyjściowym j w czasie t .

$b_{ij}(t)$ = waga połączeń z dołu do góry pomiędzy węzłem wejściowym i , a węzłem wyjściowym j w czasie t .

Wagi definiują cechy wzorca na wyjściach j .

ε = określa przedział tolerancji, w którym wynik dopasowania do wzorca można uznać za prawidłowy

Krok 2: Wprowadź nowy sygnał wejściowy

Krok 3: Oblicz wynik dopasowania do wzorca

$$\mu_j = \sum_{i=0}^{N-1} b_{ij}(t) x_i \quad 0 \leq j \leq M - 1$$

μ_j - wartość wyjścia węzła j , a x_i - element i wejścia, przyjmujący wartości 0 lub 1.

Krok 4: Dopasowanie do wzorca

$$\mu_j^* = \max_j \{\mu_j\}$$

Krok 5: Test dopasowania ε

$$\|X\| = \sum_{i=0}^{N-1} x_i$$

$$\|T \cdot X\| = \sum_{i=0}^{N-1} t_{ij} x_i$$

$$\frac{\|T \cdot X\|}{\|X\|} > \varepsilon$$

NIE

TAK

IDŹ DO KROKU 6

IDŹ DO KROKU 7

Krok 6: Unieważnienie najlepszego dopasowania

Chwilowo wynik dopasowania do wzorca w kroku 4 jest zerowany i nie bierze udziału w „maksymalizacji”
W kroku 4. Następnie skok do kroku 3.

Krok 7: Adaptacja danych najbardziej pasującego wzorca

$$t_{ij}(t+1) = t_{ij}(t)x_i$$

$$b_{ij}(t+1) = \frac{t_{ij}^*(t)x_i}{0.5 + \sum_{i=0}^{N-1} t_{ij}(t)x_i}$$

Krok 8: Powtórz wracając do kroku 2

INPUT	EXEMPLARS AFTER EACH INPUT.
	10 x 7 p pic 64

Przykład (Carpenter/Grossberg)

Dopasowywanie do trzech wzorców: liter C, E i F. Każdy wzorzec składa się z 64 pikseli.

$$\varepsilon < 0.9$$

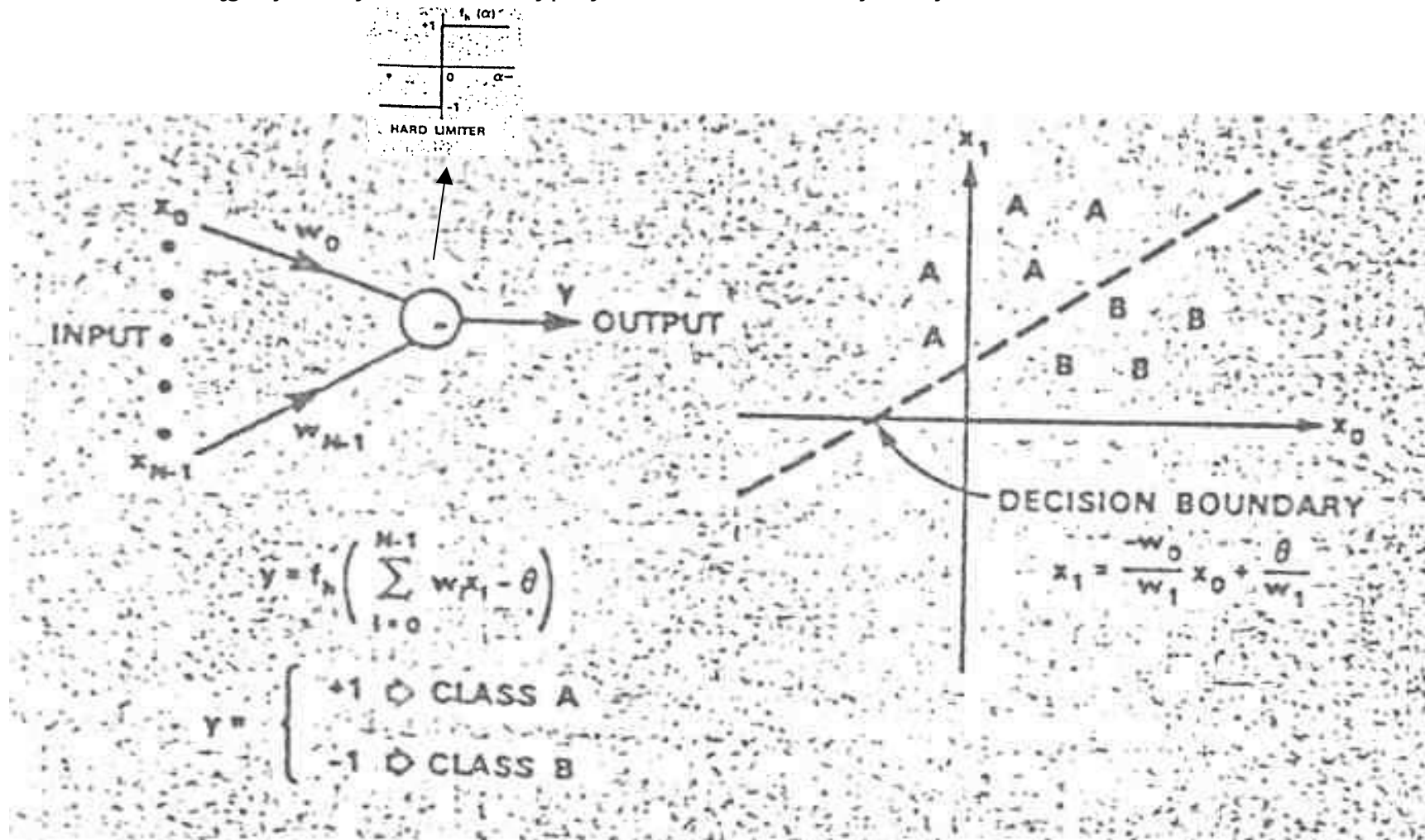
Wada: dość silny wpływ zakłóceń (szumów) na wynik.

Zaleta: prosta topografia sieci.

PERCEPTRON JEDNOWARSTWOWY JAKO KLASYFIKATOR

(R. Rosenblatt, „Principles of Neurodynamics”, New York, Spartan Books, 1959)

Wejście zarówno ciągłe jak i dyskretne. Przypisywanie elementów jednej z dwóch klas.



Wagi perceptronu modyfikowane w czasie uczenia według wielu różnych algorytmów

Oryginalny algorytm uczenia perceptronu (Rosenblatt)

Krok 1: Wyznaczenie wstępne wag i przesunięć argumentu (threshold)

Podstaw małe wartości (bliskie zera) wag, tzn. $w_i(0) \approx 0, \theta_i \approx 0 \quad (0 \leq i \leq N-1)$

Krok 2: Wprowadź nowy sygnał wejściowy (ciągły) i oczekiwana wartość wyjścia

Krok 3: Oblicz aktualną wartość wyjścia:

$$y(t) = f_h \left(\sum_{i=0}^{N-1} w_i(t) x_i(t) - \theta_i \right)$$

Krok 4: Adaptacja wag:

$$w_i(t+1) = w_i(t) + \eta [d(t) - y(t)] x_i(t), \quad 0 \leq i \leq N-1$$

$$d(t) = +1 \text{ dla przypadków klasy } A, \quad -1 \text{ dla przypadków klasy } B$$

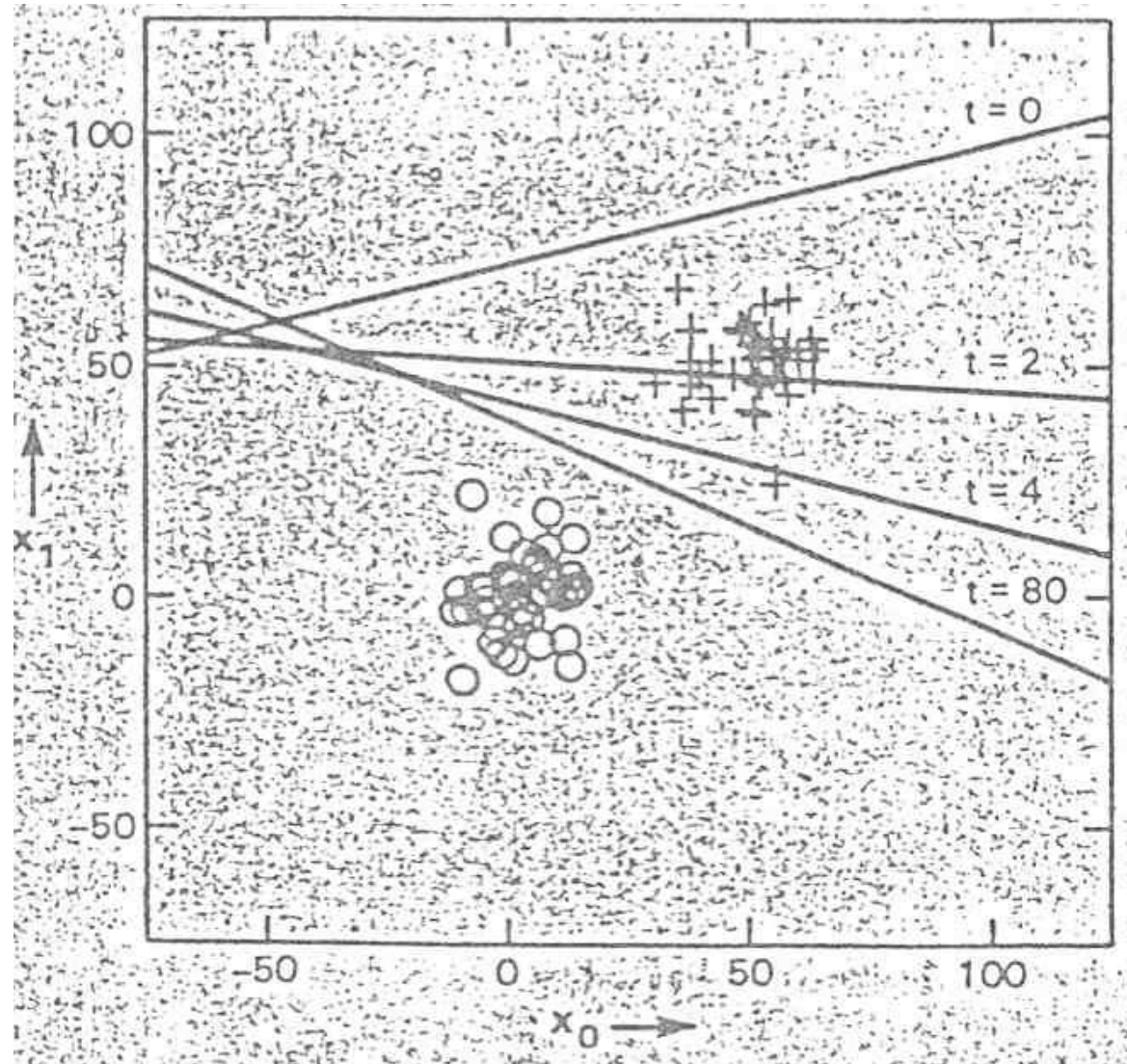
η - dodatni współczynnik wzmocnienia mniejszy od 1

$d(t)$ – oczekiwana odpowiedź

Uwaga: wagi nie są modyfikowane gdy na wyjściu pojawia się oczekiwana odpowiedź.

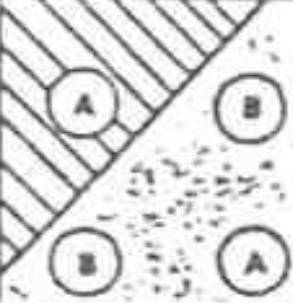
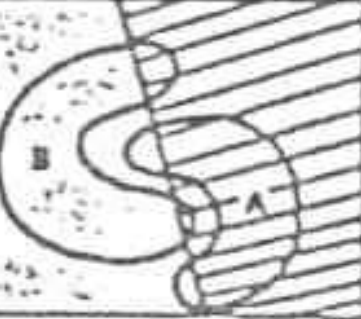
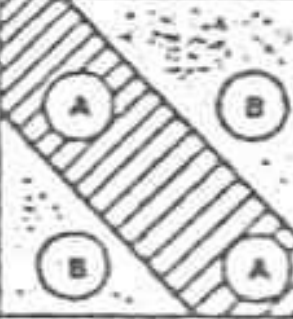
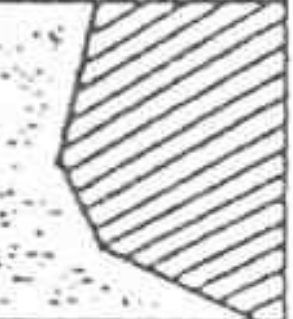
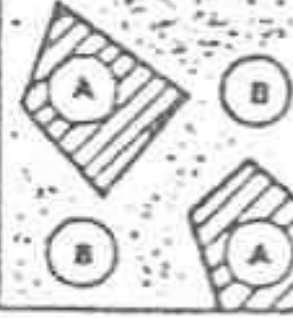
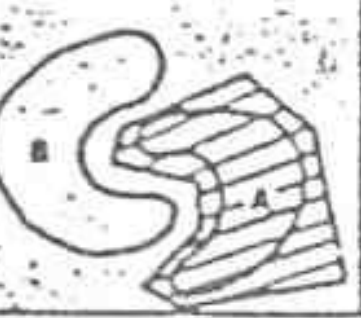
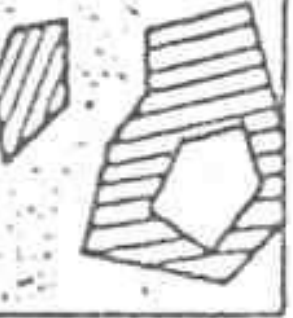
Krok 5: Powtórz krok 2 itd.

Przykład

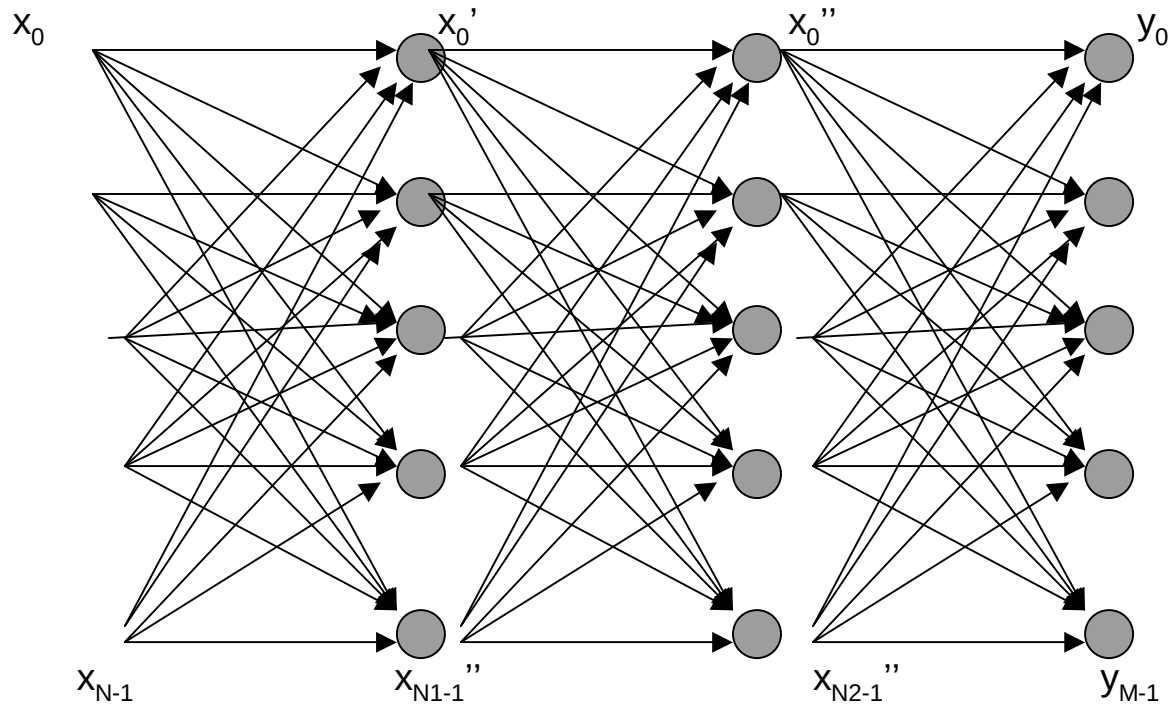


Porządek wprowadzony przez dwie klasy (kółka i krzyżyki). Niezbędna liczba iteracji = 80

TYPY REGIONÓW DECYZYJNYCH W ZALEŻNOŚCI OD TOPOGRAFII PERCEPTRONU

STRUCTURE	TYPES OF DECISION REGIONS	EXCLUSIVE OR PROBLEM	CLASSES WITH MESHED REGIONS	MOST GENERAL REGION SHAPES
SINGLE-LAYER 	HALF PLANE BOUNDED BY HYPERPLANE			
TWO-LAYER 	CONVEX OPEN OR CLOSED REGIONS			
THREE-LAYER 	ARBITRARY (Complexity Limited By Number of Nodes)			

PERCEPTRON WIELOWARSTWOWY JAKO KLASYFIKATOR



$$x'_j = f\left(\sum_{i=0}^{N-1} w_{ij} x_i - \theta_j\right) \quad 0 \leq j \leq N_1 - 1$$

$$x''_k = f\left(\sum_{j=0}^{N_1-1} w'_{jk} x'_j - \theta'_k\right) \quad 0 \leq k \leq N_2 - 1$$

$$y_l = f\left(\sum_{k=0}^{N_2-1} w''_{kl} x''_k - \theta''_l\right) \quad 0 \leq l \leq M - 1$$

Algorytm uczenia sieci „back propagation”

Neurony posiadają charakterystyki sigmoidalne, tzn.:

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha - \beta)}}$$

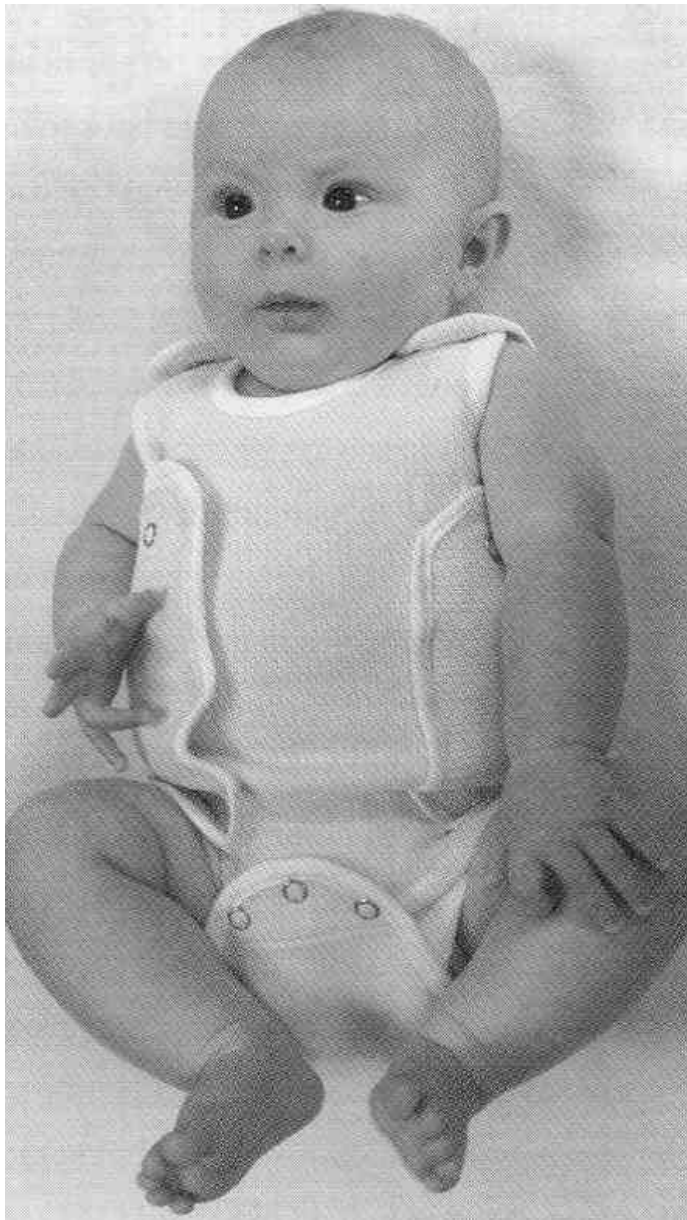
Krok 1. Inicjalizacja wag i offsetów (Θ)

INTELIGENTNA ODZIEŻ

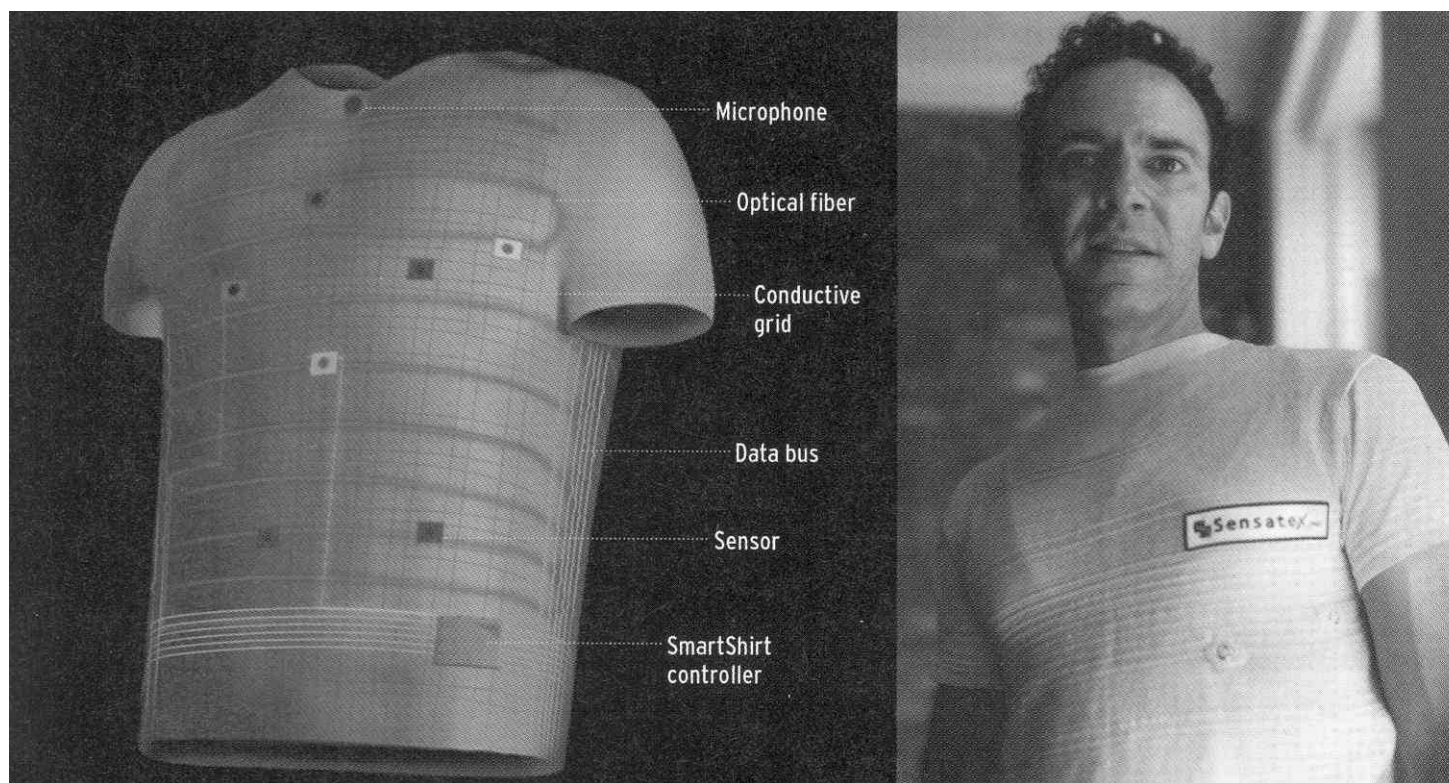


Inteligentne
czujniki

Nadajniki, odbiorniki GPS
Źródła zasilania
Procesory i pamięci



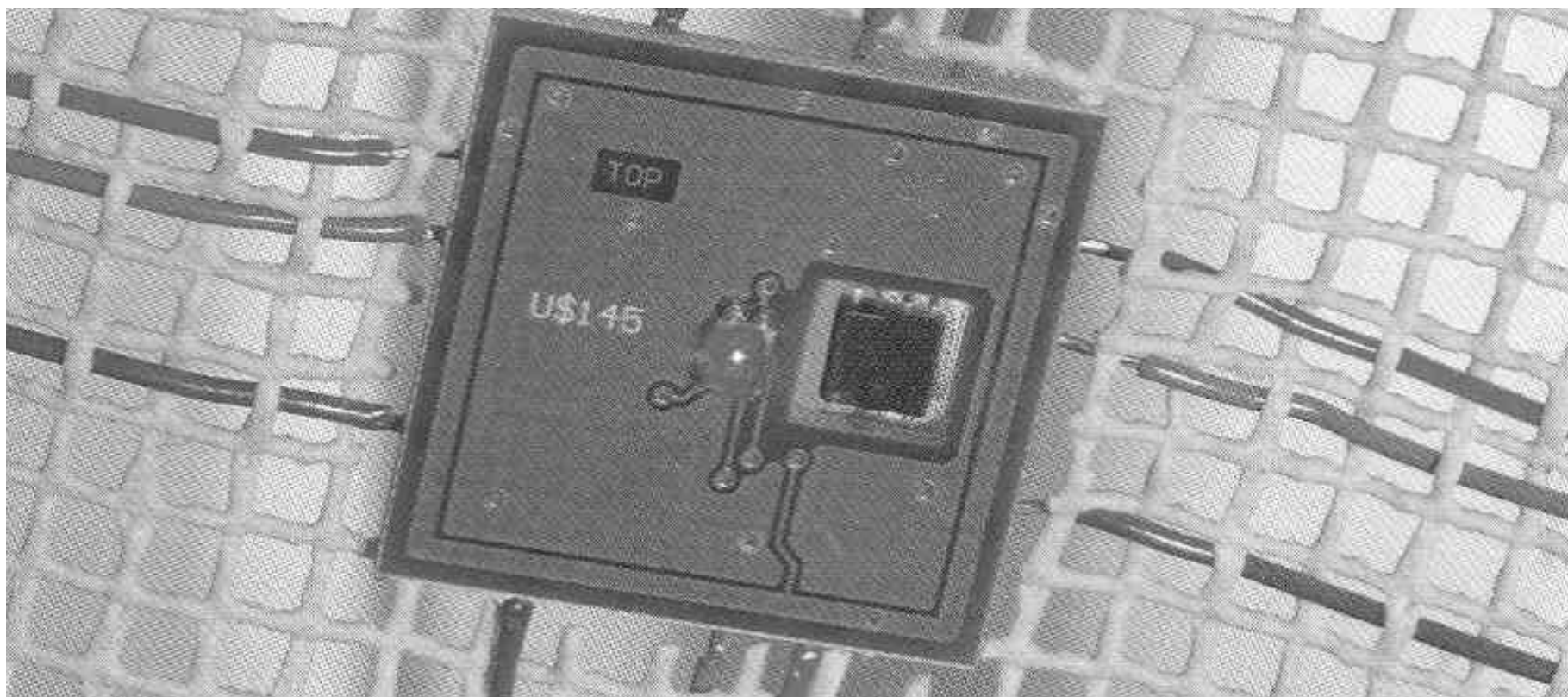
?



ODZIEŻ WYCZYNOWA

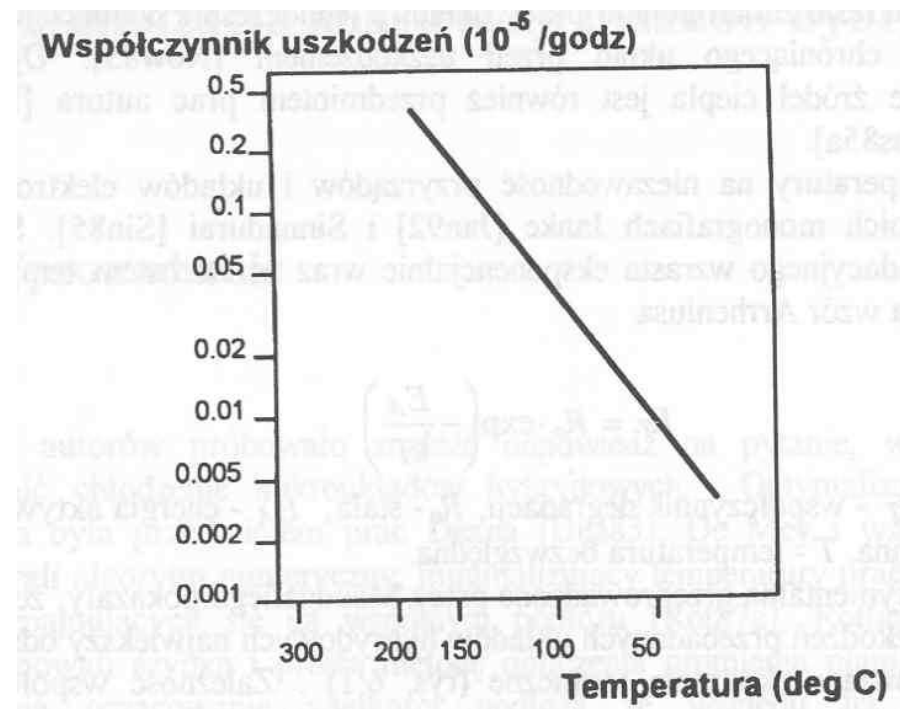
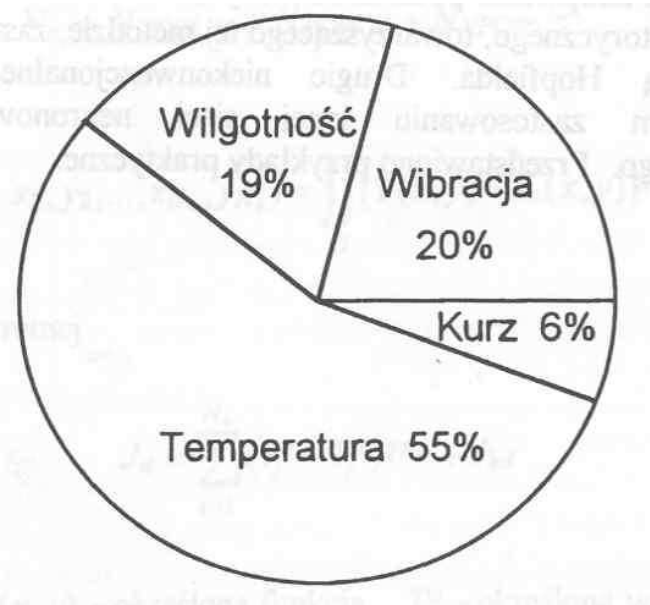


Coś dla rowerzystów



„Inteligentna” tkanina (SoC uczący się + zasilanie)

Przyczyny uszkodzeń układów i systemów mikroelektronicznych



Termiczna optymalizacja topografii układu VLSI

Postawienie problemu

$$N_e = N_{POW_N} + N_{POW_S} + N_{NPOW_S}$$

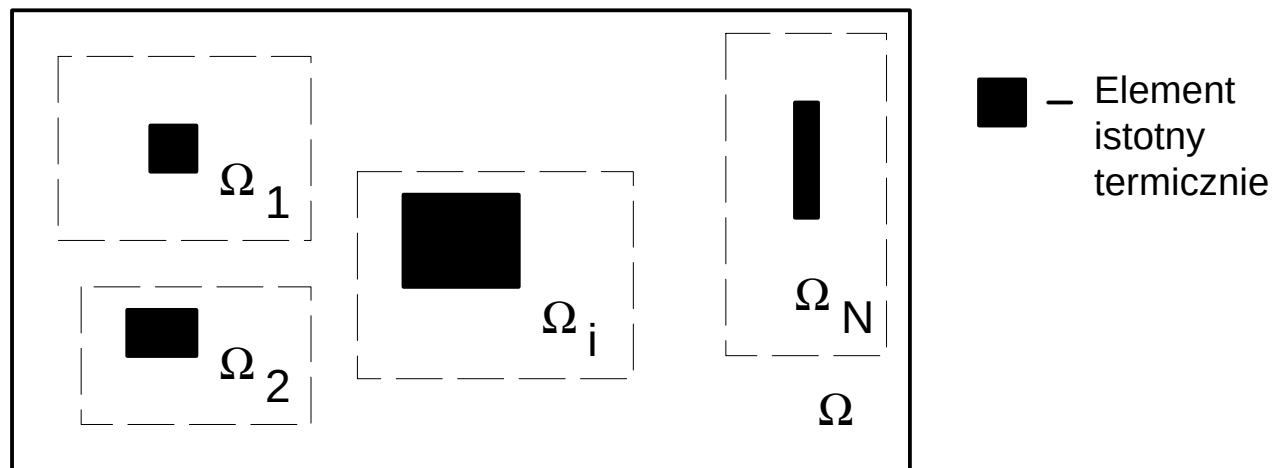
Przykładowa funkcja celu

$$J_d = \sum_{i=1}^{N_e} (T_i^c - T_i^o)^{2N_e} + J_{kd}$$

przy czym $T_o(x, y)$ - określona funkcja, T_i^o - określona wartość,
 T_i^c - temperatura obliczona w środku i -tego elementu,
 $J_k, (J_{kd})$ - funkcja (wartość) kary za przekroczenie dopuszczalnej temperatury.

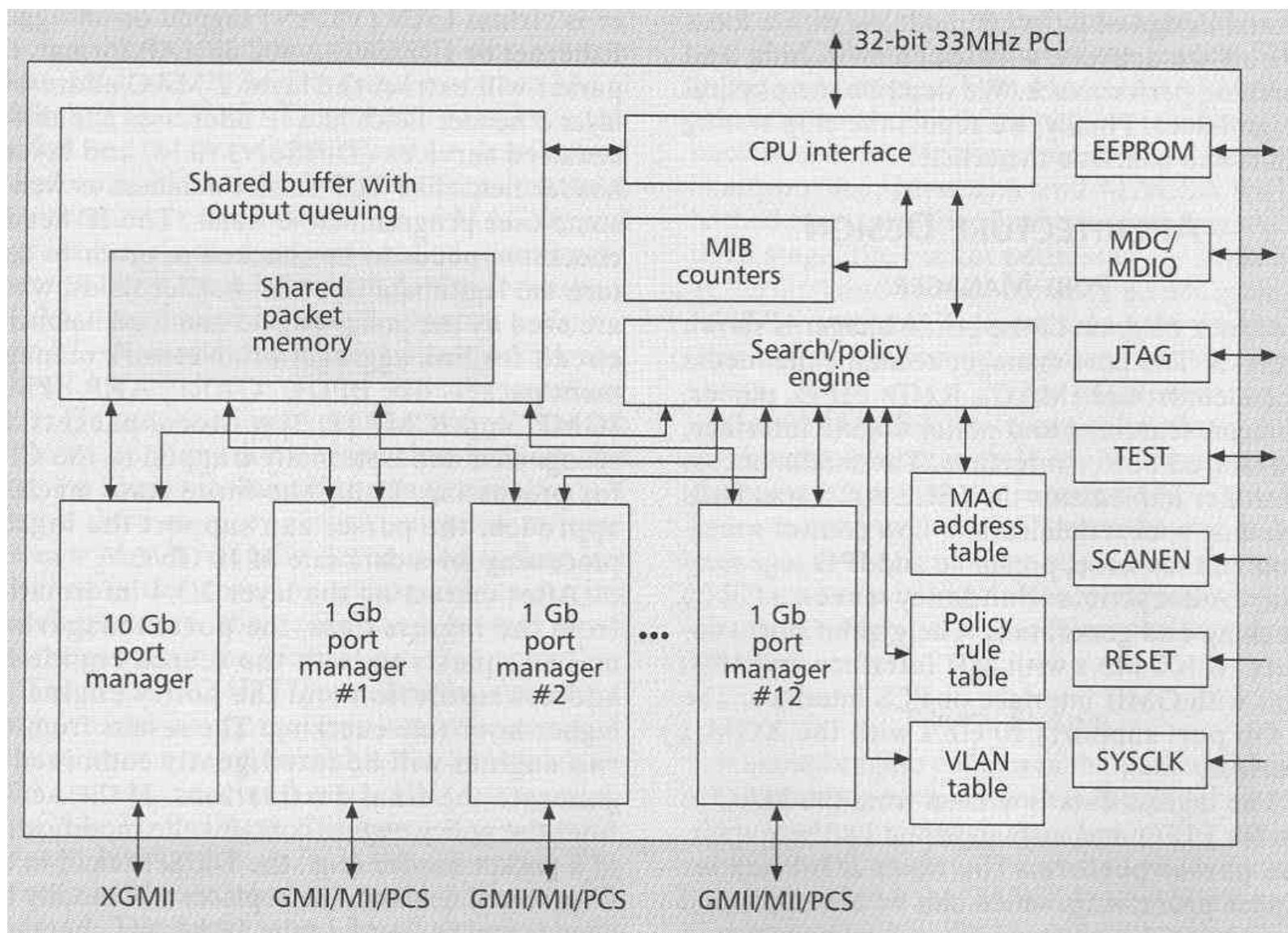
Ze względu na wybór $T_o(x, y)$ mamy następujące przypadki funkcji celu:

- a) $T_o(x, y) = const = T_m$. Minimalizacja odchyłki od temperatury średniej.
To kryterium prowadzi do rozkładu temperatury podłoża zbliżonego do równomiernego, (minimalizacja naprężeń termomechanicznych w elementach znajdujących się na podłożu).
- b) $T_o(x, y) = const = T_a$. Minimalizacja bezwzględnej temperatury pracy każdego elementu istotnego termicznie.
- c) $T_o(x, y)$ - z góry określony rozkład temperatury na podłożu. Kryterium prowadzi do minimalizacji odchyłki od zadanego pola temperatury.

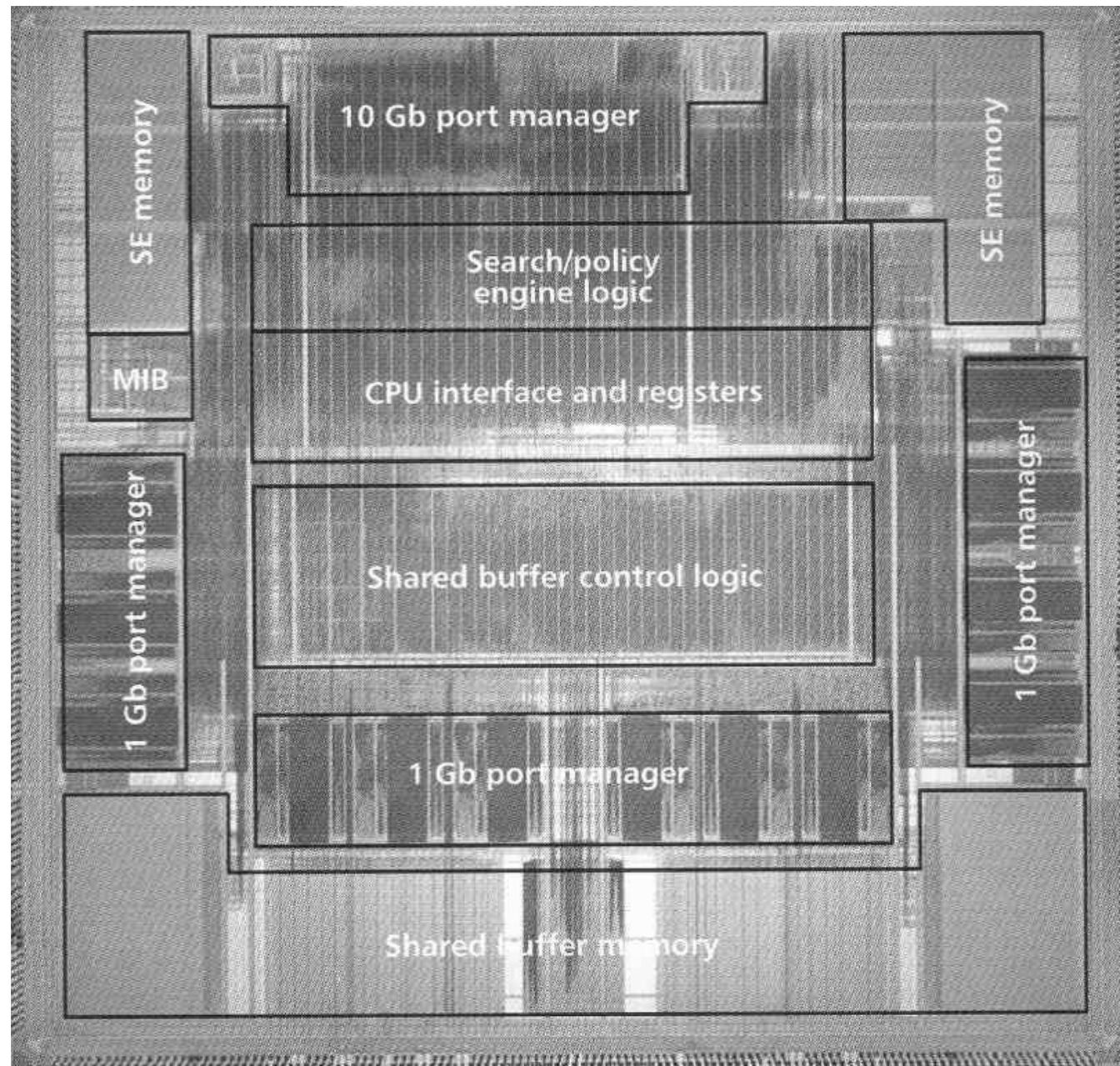


Przykładowe rozmieszczenie podobszarów Ω_i na podłożu

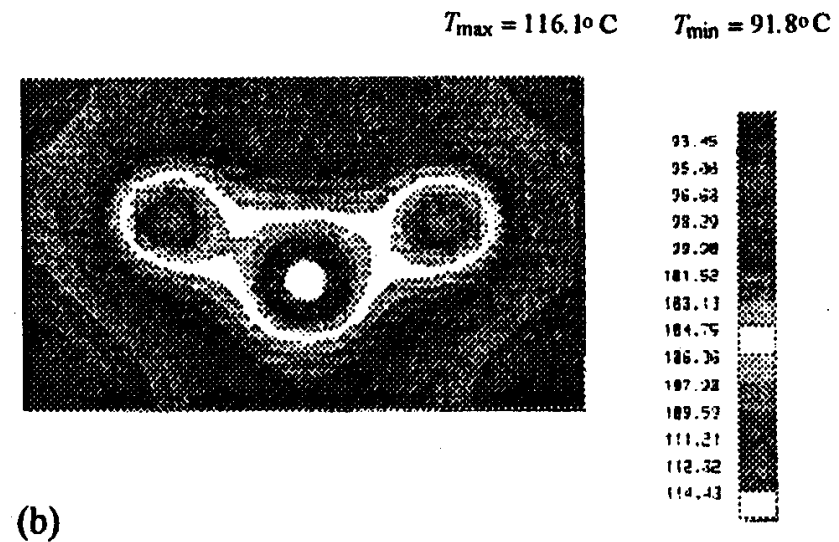
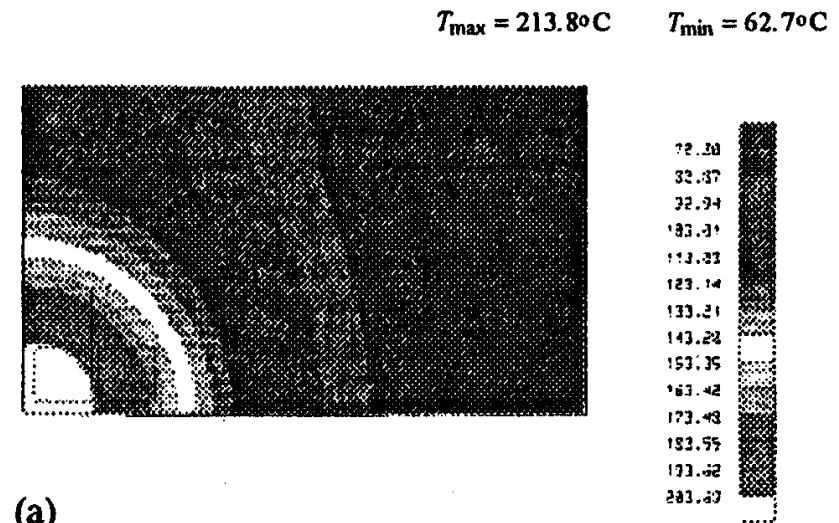
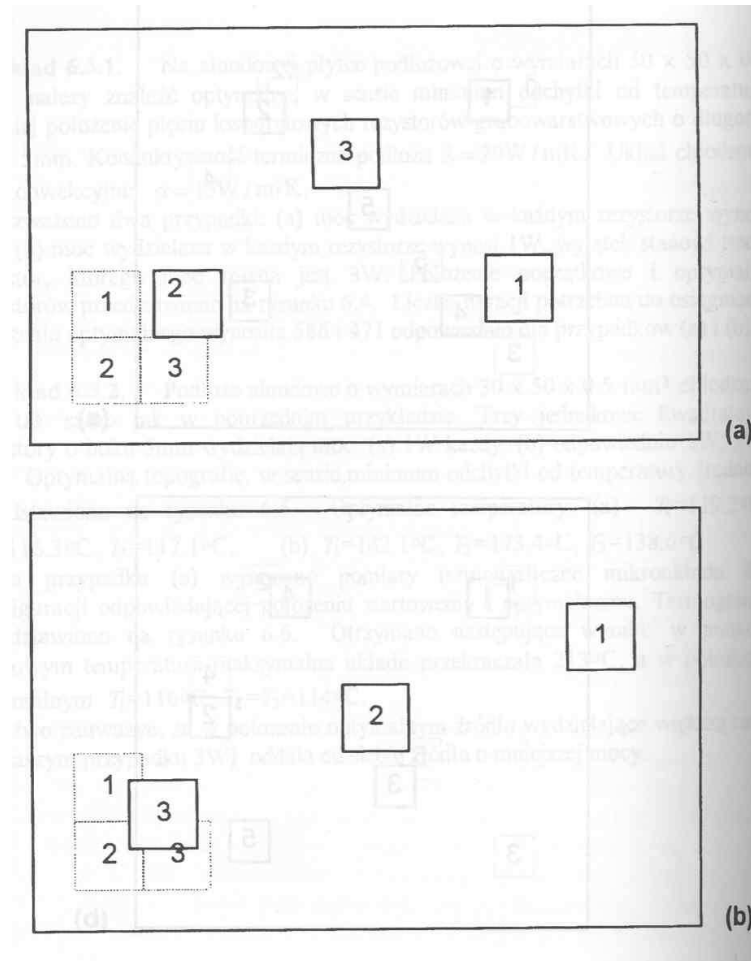
Przykładowa topografia wybranego procesora



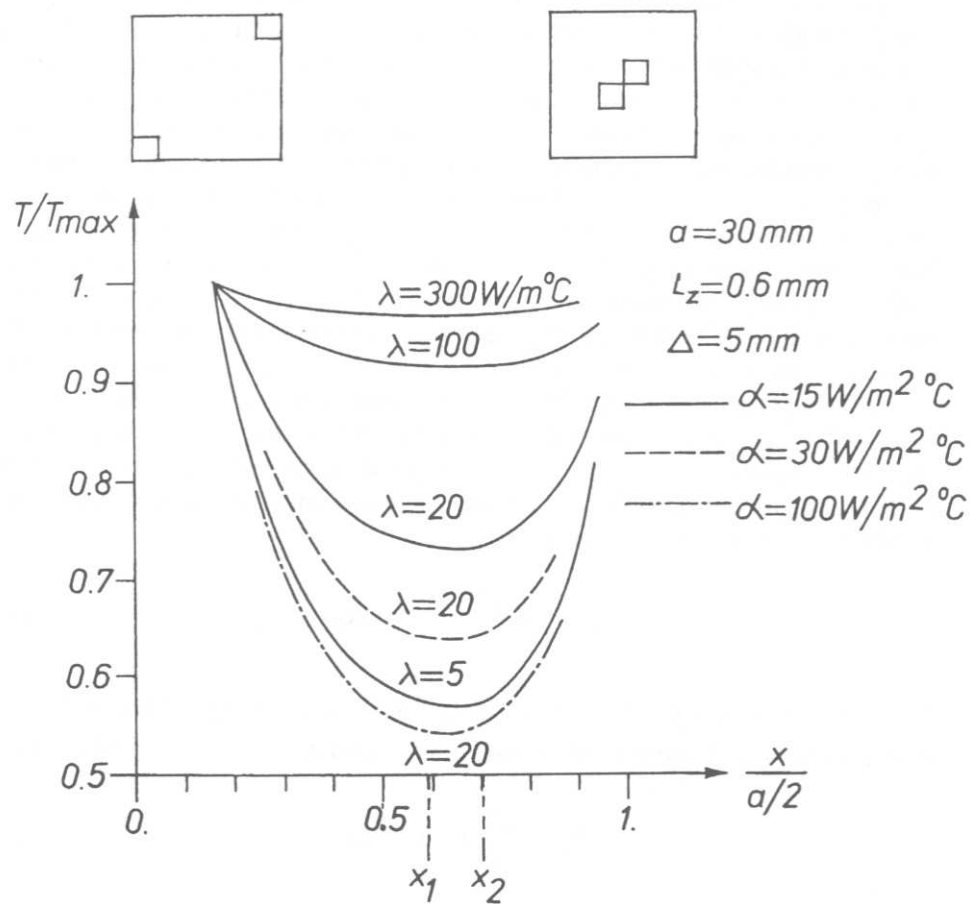
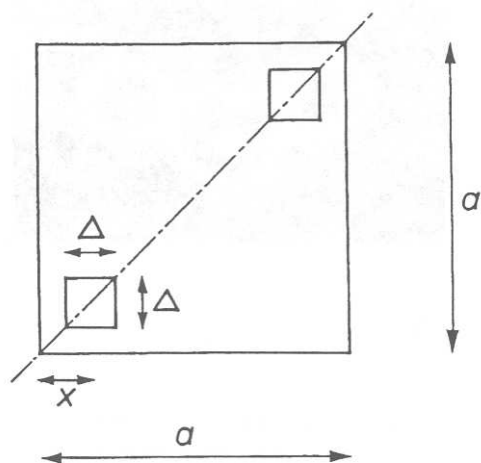
Fotografia modułu scalonego



Klasyczne metody optymalizacji

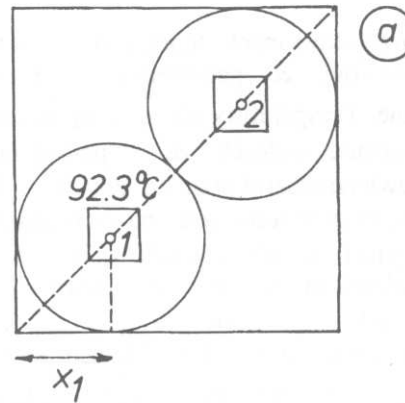


Metoda heurystyczna rozmieszczenia źródeł ciepła na wspólnym podłożu

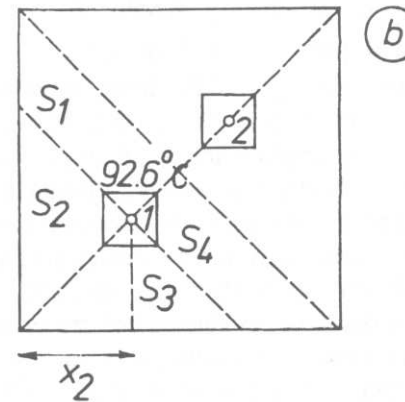


Podział podłoża na symetryczne obszary (problem 2 źródeł)

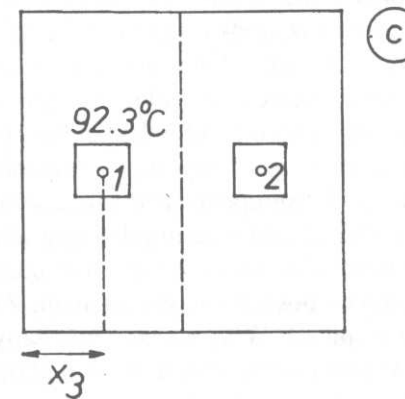
$$\begin{aligned}
 P_1 &= P_2 = 1 \text{ Watt} \\
 \lambda &= 20 \text{ W/m}^\circ\text{C} \\
 \alpha &= 15 \text{ W/m}^2 \text{ }^\circ\text{C} \\
 a &= 30 \text{ mm} \\
 L_z &= 0.6 \text{ mm} \\
 x_1 &= \frac{2-\sqrt{2}}{2} a
 \end{aligned}$$

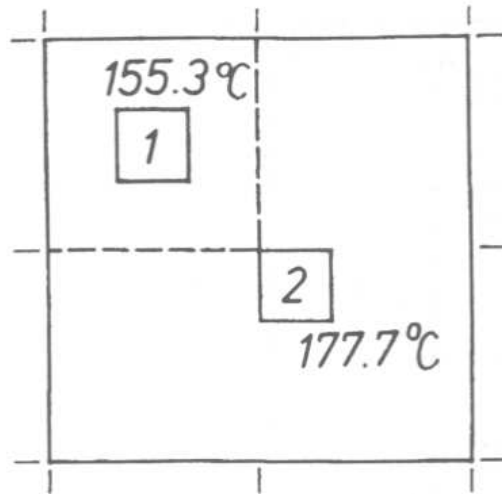


$$x_2 = \frac{\sqrt{2}}{4} a$$



$$x_3 = \frac{1}{4} a$$





**Przypisanie źródłom obszarów
proporcjonalnych do ich mocy strat**

$$P_2 = 3P_1 \quad P_1 = 0.5 \text{ Watt}$$

$$\lambda = 20 \text{ W/m}^\circ\text{C}$$

$$\alpha = 15 \text{ W/m}^2 \text{ }^\circ\text{C}$$

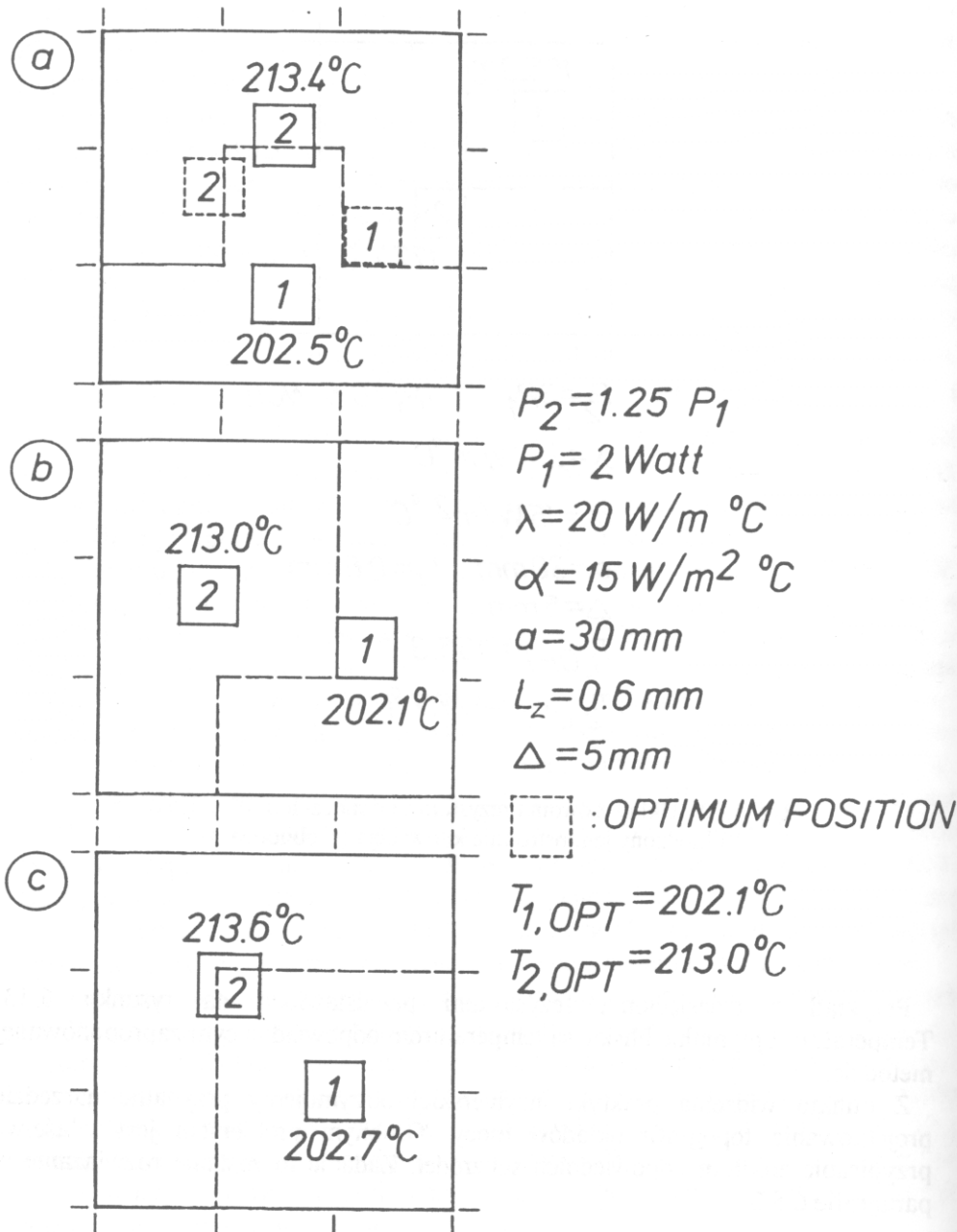
$$a = 30 \text{ mm} \quad L_z = 0.6 \text{ mm}$$

$$\Delta = 5 \text{ mm}$$

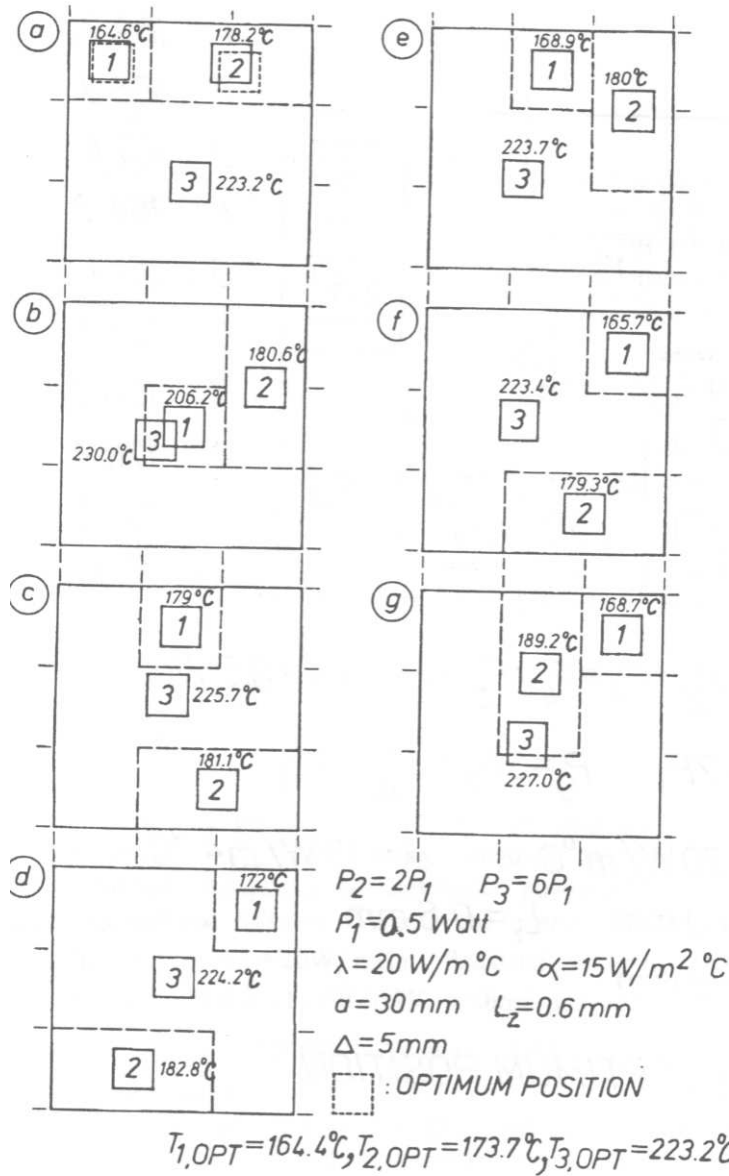
$$T_{1,OPT} = 155.3^\circ\text{C}$$

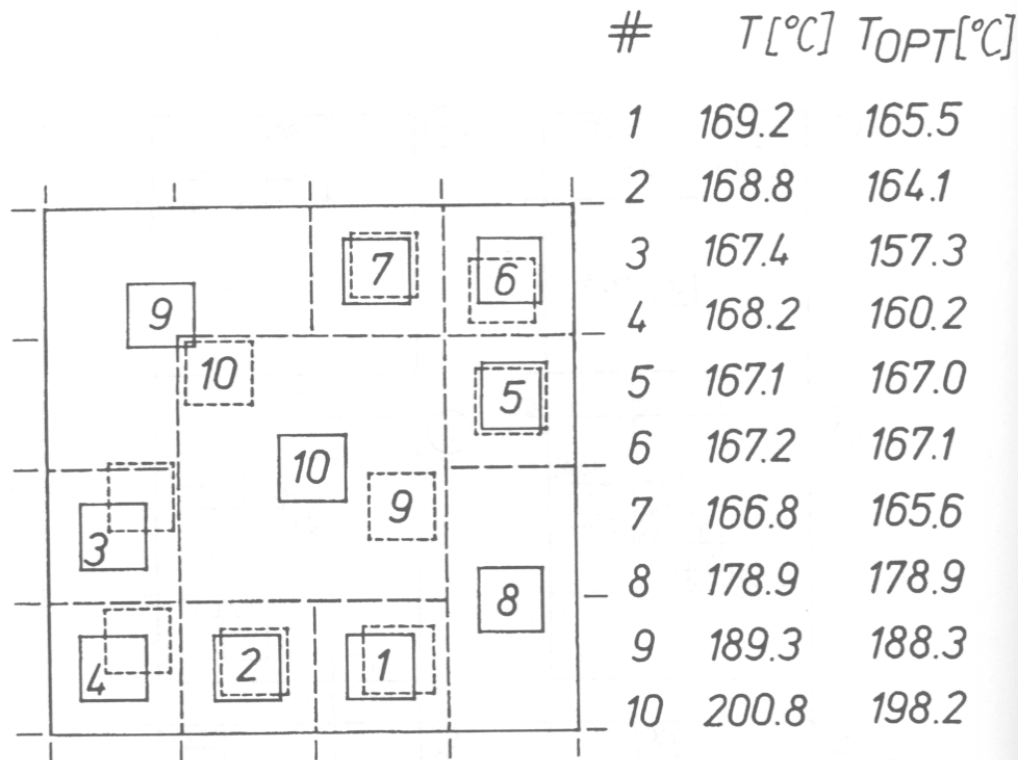
$$T_{2,OPT} = 177.7^\circ\text{C}$$

**Wielowariantowe rozwiązanie
problemu 2 źródeł**



Wielowariantowe rozwiązanie Wybranego problemu 3 źródeł





Układ z 10 źródłami

$$P_1 = P_2 = P_3 = P_4 = P_5 = P_6 = P_7 = 0.5 \text{ Watt}$$

$$P_8 = 2P_1 \quad P_9 = 3P_1 \quad P_{10} = 4P_1$$

$$\lambda = 20 \text{ W/m}^\circ\text{C} \quad \alpha = 15 \text{ W/m}^2 \text{ }^\circ\text{C}$$

$$a = 40 \text{ mm} \quad L_z = 0.6 \text{ mm}$$

$$\Delta = 5 \text{ mm}$$

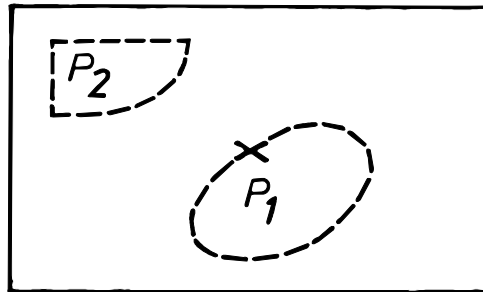
 : OPTIMUM POSITION

Optymalizacja topografii przy pomocy sieci neuronowych

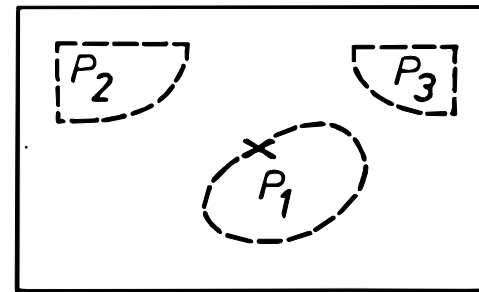
Założenia wstępne:

Propozycja punktu startowego – obszary charakterystyczne

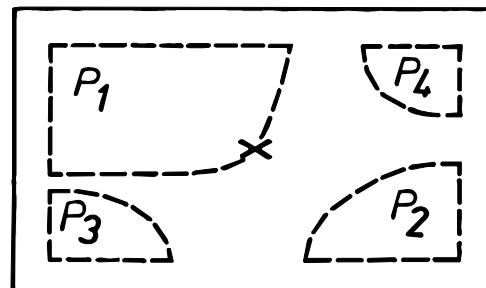
2 heat sources



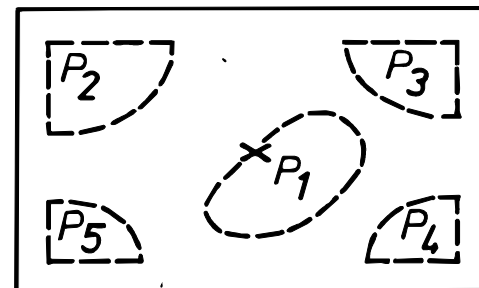
3 heat sources



4 heat sources



5 heat sources



Sieć neuronowa składa się z $N+1$ subsieci. Każda subsieć jest wielowarstwowym perceptronem. Pierwsza warstwa utworzona jest przez węzły, które zwielokrotniają sygnały wejściowe bez zmiany ich wartości, $f(\vartheta) = \vartheta$. Warstwa ukryta i wyjściowa składa się z neuronów, to znaczy elementów realizujących funkcję

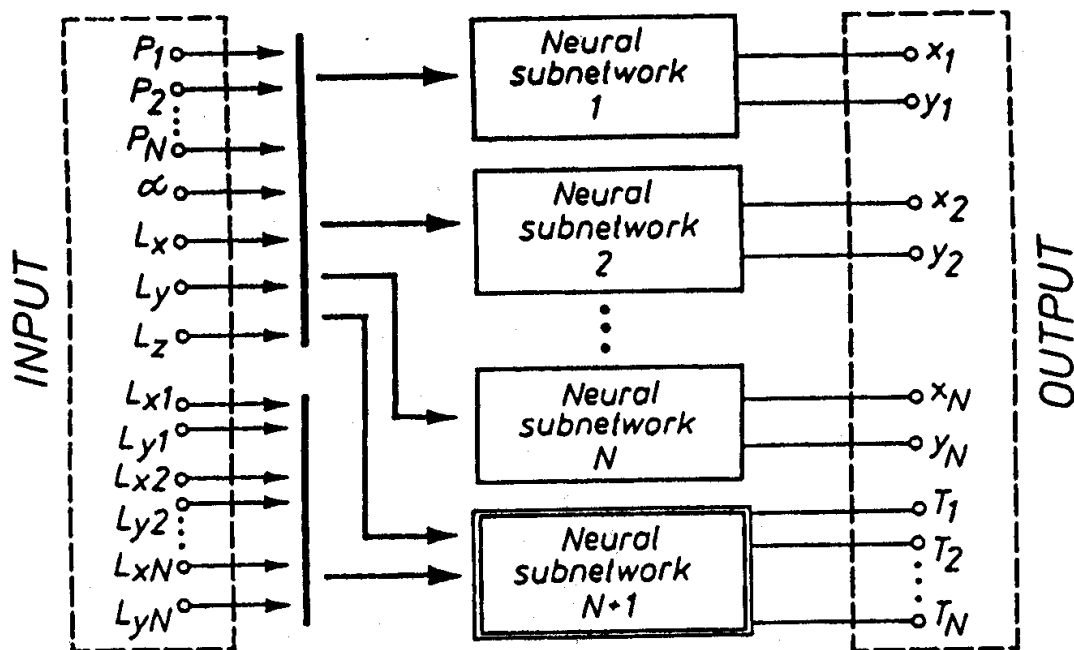
$$f(\theta) = \frac{1}{1 + \exp(-\theta - \phi)}$$

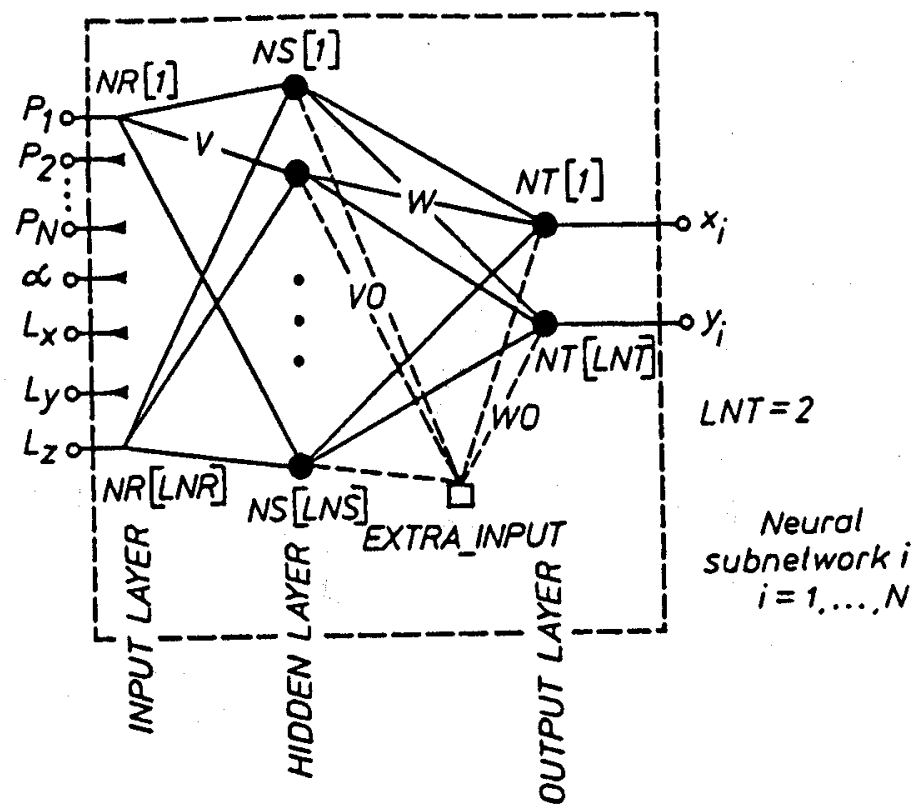
przy czym θ jest sumą algebraiczną sygnałów wchodzących do neuronu, ϕ reprezentuje istnienie dodatkowego wejścia *EXTRA_INPUT* równego 1.

Subsieci o numerach od 1 do N przeznaczone są do generowania optymalnej topografii, zaś subsieć $N+1$ do estymacji temperatury w położeniu optymalnym. Oszacowanie temperatur możliwe jest dzięki spostrzeżeniu, że

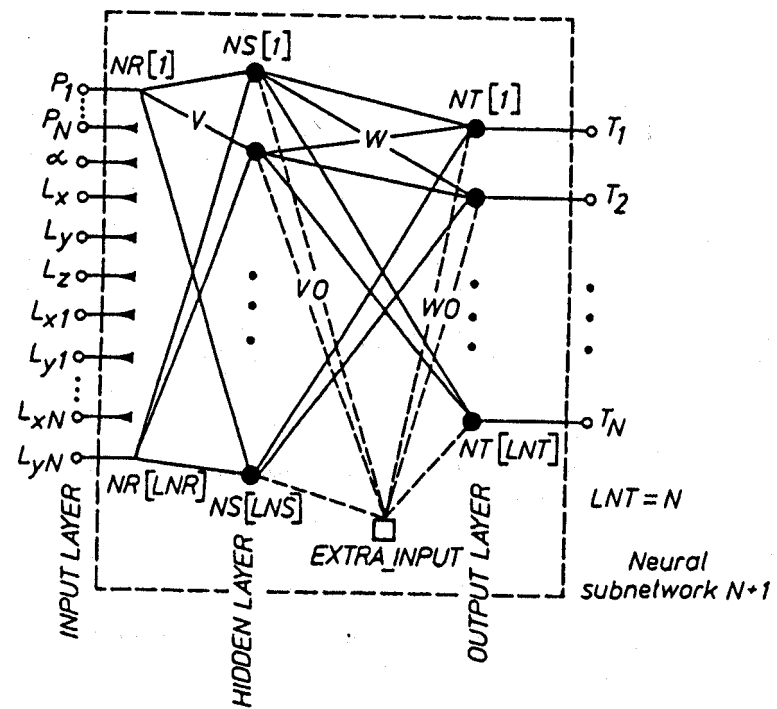
$$T_i = T_m + \Delta T_i, \quad i = 1, \dots, N$$

przy czym T_m - temperatura średnia

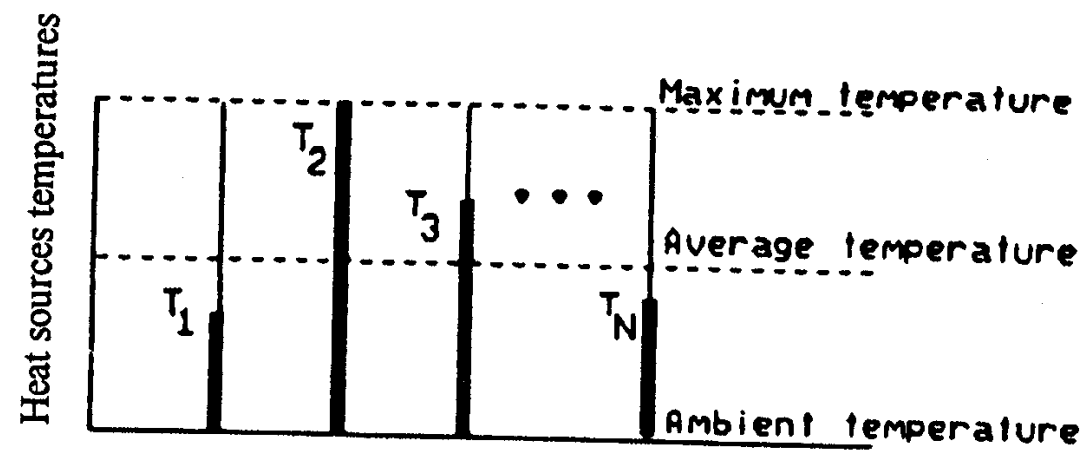




Subsieć wyznaczająca optymalne położenie i -tego źródła ciepła



Subsieć do oszacowania temperatur w położeniu optymalnym



Temperatury poszczególnych źródeł ciepła

Wielkościami wejściowymi kompleksowej sieci neuronowej są: $P_1, P_2, \dots, P_N$ -moce źródeł ciepła, α - współczynnik konwekcji, L_x, L_y, L_z - wymiary podłoża, Δ_{xi}, Δ_{yi} - wymiary źródeł ($i=1, \dots, N$). Konduktywność podłoża ma wartość ustaloną. Wszystkie wielkości są skalowane w taki sposób, że ich wartości mieszczą się w przedziale od 0 do 1. Wagi sieci korygowane są w czasie uczenia według oryginalnej reguły: dla $i=1, \dots, LNS$ i $j=1, \dots, LNT$

$$\begin{aligned} w[i, j](t+1) &= w[i, j](t) + \eta \cdot NS[i] \cdot d[j] \\ wo[j](t+1) &= wo[j](t) + \eta \cdot EXTRA_INPUT \cdot d[j] \end{aligned}$$

przy czym

$$d[j] = NT[j]_{opt} - NT[j]$$

Dla $i=1, \dots, LNR$ i $j=1, \dots, LNS$

$$\begin{aligned} v[i, j](t+1) &= v[i, j](t) + \eta \cdot \theta[j] \cdot NR[i] \\ vo[j](t+1) &= vo[j](t) + \eta \cdot \theta[j] \cdot EXTRA_INPUT \end{aligned}$$

przy czym

$$0 < \eta < 1$$

$$\theta[j] = \sum_{k=1}^{LNT} w[j, k] \cdot d[k]$$

Algorytm "back-propagation" minimalizuje różnicę pomiędzy aktualnym wyjściem $NT[i]$ perceptronu, a żadaną wartością optymalną, to znaczy $NT[i]_{opt} = x[i]_{opt}, y[i]_{opt}$ lub $T[i]_{opt}$. Procedura uczenia powtarzana jest $ITER$ razy. Wartości optymalne obliczono przy pomocy metody gradientowej. Dla oceny dokładności nauczonego perceptronu wprowadzono następujące kryteria:

- Błąd średniokwadratowy położenia

$$\delta_{geo} = \sqrt{\frac{\sum_{i=1}^{ITER_TEST} \sum_{j=1}^{2N} (NT[j]_{opt} - NT[j])^2}{2N \cdot ITER_TEST}}$$

- Błąd średniokwadratowy temperatury

$$\delta_{temp} = \sqrt{\frac{\sum_{i=1}^{ITER_TEST} \sum_{j=1}^{2N} (T[j]_{opt} - T[j])^2}{N \cdot ITER_TEST}}$$

- Maksymalny błąd położenia

$$\delta_{geo\ max} = \max_{i=1, \dots, 2N, 1, \dots, ITER_TEST} ABS(NT[i]_{opt} - NT[i])$$

- Maksymalny błąd temperatury $\delta_{temp\ max}$ - analogiczny do błędu położenia, przy czym $ITER_TEST$ jest liczbą cykli testujących nauczoną sieć.

Przykład 1. Problem polega na optymalnym rozmieszczeniu, w sensie minimum temperatury źródeł, pięciu źródeł na podłożu alundowym ($\lambda = 20 \text{ W/mK}$) o wymiarach $50 \times 50 \times 0.625 \text{ mm}^3$. Układ chłodzony jest dwustronnie konwekcyjnie, $15 \leq \alpha \leq 100 \text{ W/m}^2\text{K}$. Moce źródeł $1 \text{ W} \geq P_1 \geq \dots \geq P_5 \geq 0$. Wymiary źródeł $\Delta_{xi}, \Delta_{yi} = 5 \text{ mm}$, dla $i = 1, \dots, 5$. Liczba cykli uczenia $ITER = 10000$. W każdym cyklu dane wejściowe generowano losowo z równomiernym rozkładem prawdopodobieństwa. Nauczoną sieć wykorzystano do generacji optymalnych topografii dla $ITER_TEST = 1000$ losowo wybranych przykładów danych wejściowych. Otrzymano następujące wartości kryteriów:

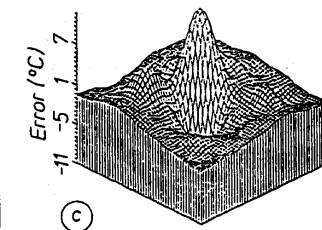
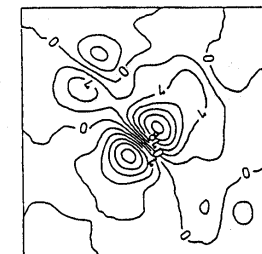
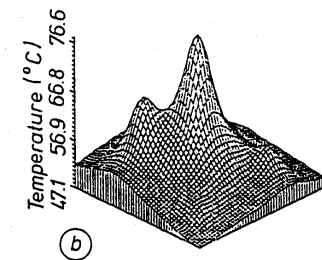
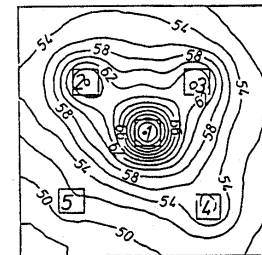
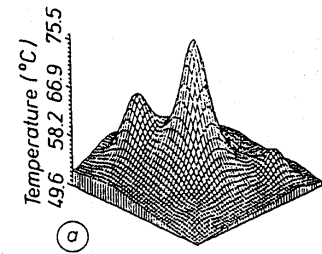
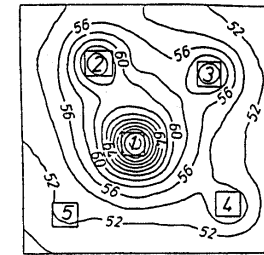
$\delta_{geo} = 1.4 \text{ mm}$, $\delta_{temp} = 3.1^\circ \text{C}$, $\delta_{geomax} = 3.2 \text{ mm}$, $\delta_{tempmax} = 12.4^\circ \text{C}$ (= 18.6% w odniesieniu do temperatury optymalnej).

Rysunek porównuje przykładową topografię wygenerowaną przez sieć neuronową z topografią optymalną, otrzymaną metodą gradientową. Dane:

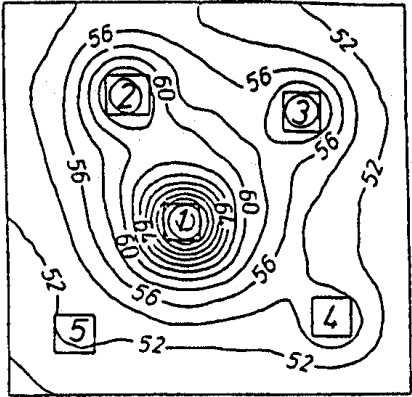
$P_1 = 1, P_2 = 0.5, P_3 = 0.4, P_4 = 0.25, P_5 = 0.1 \text{ W}$, $\alpha = 15 \text{ W/m}^2\text{K}$, $T_a = 25^\circ \text{C}$. Wyniki:

$T_{1,opt} = 76.8^\circ \text{C}$, $T_{2,opt} = 65^\circ \text{C}$, $T_{3,opt} = 59.1^\circ \text{C}$, $T_{4,opt} = 55.8^\circ \text{C}$, $T_{5,opt} = 52.4^\circ \text{C}$.

$T_{1,nn} = 78.1^\circ \text{C}$, $T_{2,nn} = 66.1^\circ \text{C}$, $T_{3,nn} = 63.4^\circ \text{C}$, $T_{4,nn} = 56.2^\circ \text{C}$, $T_{5,nn} = 53.1^\circ \text{C}$.

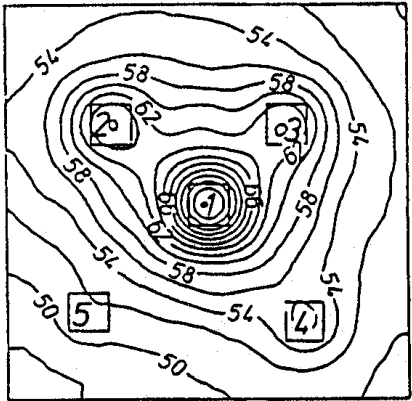
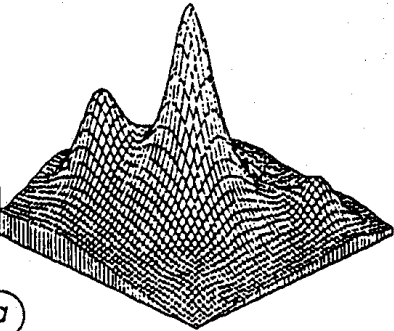


Do przykładu 1



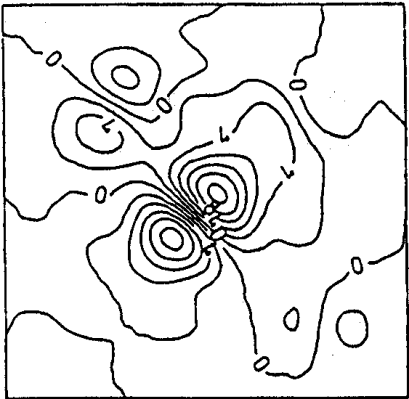
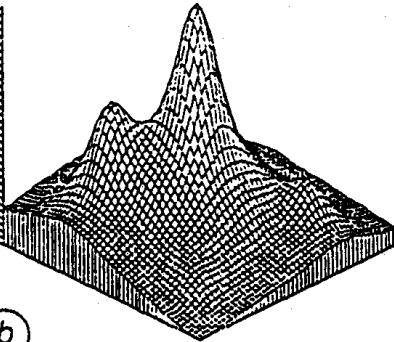
Temperature ($^{\circ}\text{C}$)
49.6 58.2 66.9 75.5

(a)



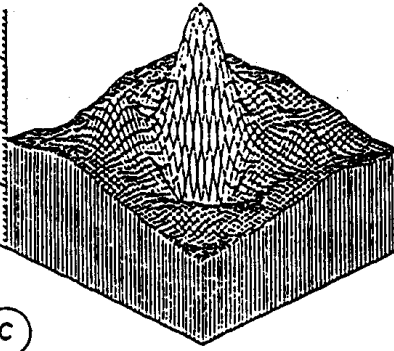
Temperature ($^{\circ}\text{C}$)
47.1 56.9 66.8 76.6

(b)



Error ($^{\circ}\text{C}$)
-11 -5 1 7

(c)



Przykład 2. Struktura sieci przedstawiona w przykładzie 1 została użyta do projektowania układów, które różnią się od poprzednich sposobem chłodzenia, wymiarami i liczbą źródeł ciepła. Tym razem minimalizowana była odchyłka temperatury pracy źródeł od temperatury średniej. Liczba źródeł wynosiła 4, sposób chłodzenia konwekcyjno-kondukcyjny, wymiary podłoża $40 \times 50 \times 0.625 \text{ mm}^3$, wymiary źródeł $\Delta_{xi} = 5\text{mm}$, $\Delta_{yi} = 6\text{mm}$, $i=1,\dots,4$. Wyniki symulacji 1000 losowych przykładów przy pomocy nauczanej sieci są następujące:

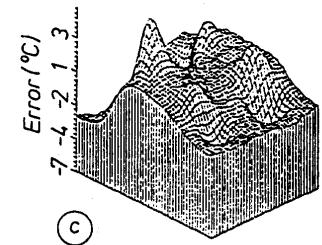
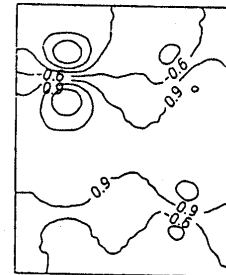
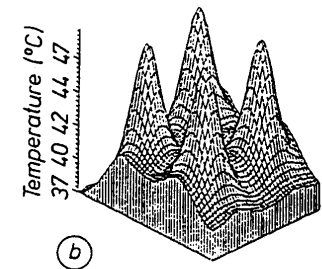
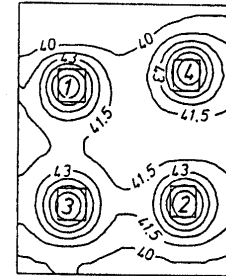
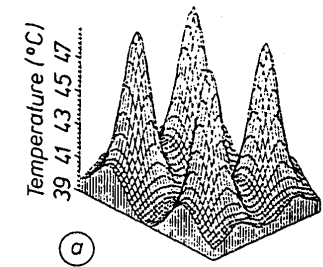
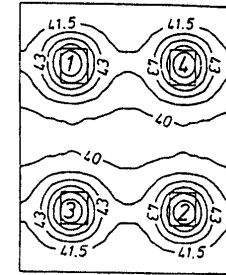
$\delta_{geo} = 1.4\text{mm}$, $\delta_{temp} = 3.6^\circ\text{C}$, $\delta_{geomax} = 2.2\text{mm}$, $\delta_{tempmax} = 2.8^\circ\text{C}$ ($=14.5\%$ w

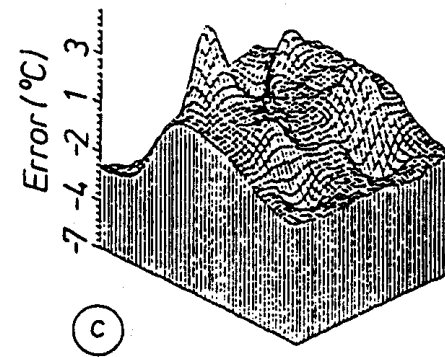
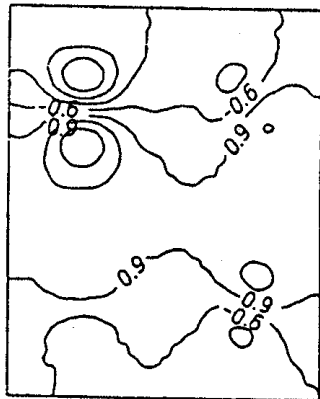
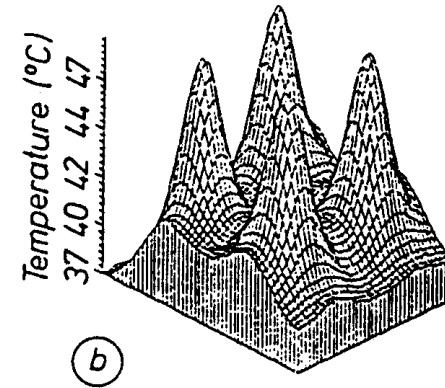
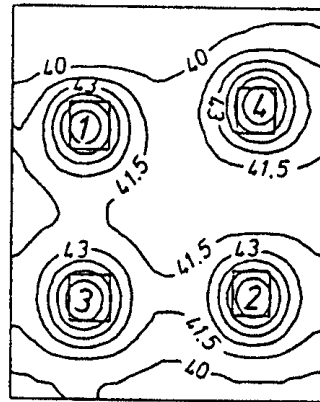
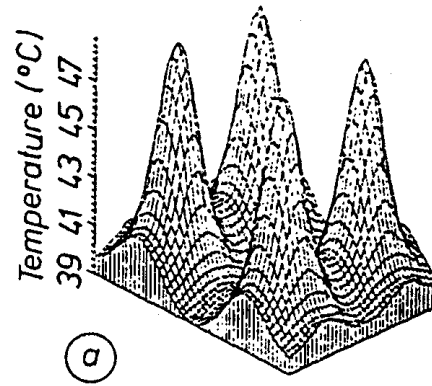
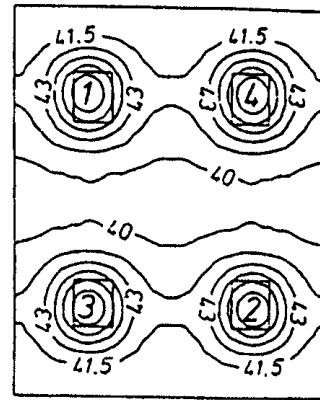
odniesieniu do temperatury optymalnej). Rysunek przedstawia przykładowy układ zaprojektowany dwójako: metodą neuronową oraz tradycyjną gradientową.

Dane: $P_1 = P_2 = P_3 = P_4 = 0.5\text{W}$, $\alpha = 15\text{W/m}^2\text{K}$, $T_c = T_a = 20^\circ\text{C}$.

Wyniki: $T_{1,opt} = T_{2,opt} = T_{3,opt} = T_{4,opt} = 48.8^\circ\text{C}$,

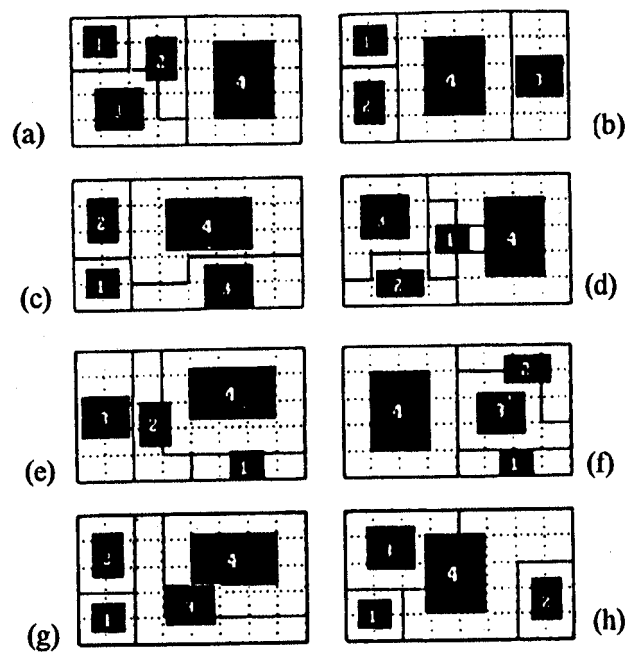
$T_{1,nn} = 48.8^\circ\text{C}$, $T_{2,nn} = 49.1^\circ\text{C}$, $T_{3,nn} = 49.1^\circ\text{C}$, $T_{4,nn} = 49.0^\circ\text{C}$.





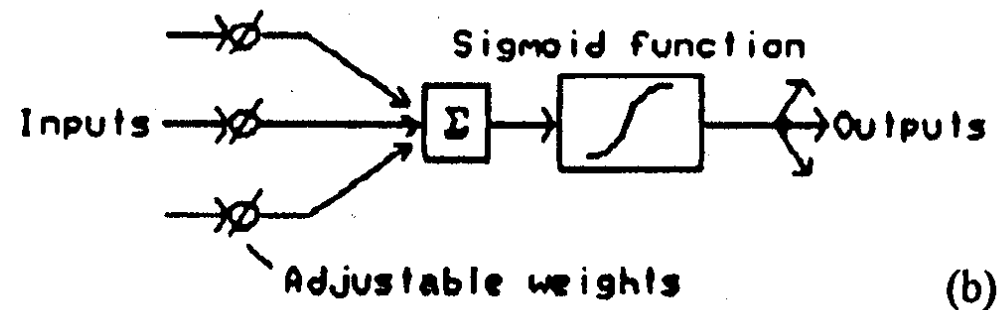
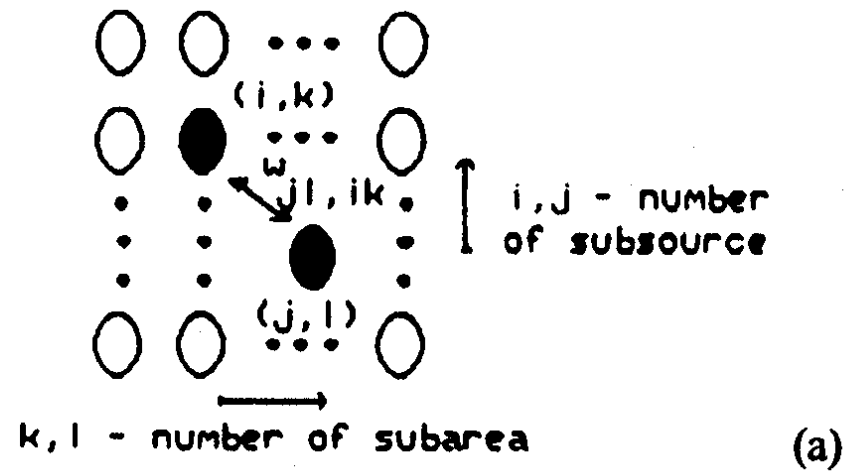
Do przykładu 2

SIEĆ HOPFIELDA



Pewne możliwości rozmieszczenia 4 źródeł ciepła na podłożu
 $P_1=0.4W$, $P_2=0.6W$, $P_3=1W$, $P_4=2W$

Sieć Hopfielda



W celu wyboru najlepszego rozwiązania w sensie minimum S użyto oryginalnie sieci neuronowej Hopfielda. W naszym przypadku minimalizacja energii sieci odpowiada minimalizacji długości obwodów S_i . Sieć przypomina macierz o wymiarach $n \times n$. Współrzędne (i, k) określają obecność i -tego subźródła w k -tej ramce. Liczba subźródeł odpowiada liczbie ramek. Każde subźródło przypisane jest jednemu i tylko jednemu źródłu.

Wagi sieci są symetryczne, to znaczy $w_{ik,jl} = w_{jl,ik}$. Nie istnieją autosprężenia, $w_{ik,ik} = 0$. Stan neuronu określony jest poziomem aktywacji u_{ik} opisanym równaniem

$$\frac{du_{ik}}{dt} = -\frac{u_{ik}}{\tau} + \sum_{j,l=1}^n w_{jl,ik} x_{jl} + I_{ik}$$

$i, k = 1, \dots, n$

przy czym: t - czas,

τ - stała czasowa,

n - liczba ramek równa liczbie subźródeł,

$w_{jl,ik}$ - waga połączenia pomiędzy elementami (j, l) i (i, k) ,

I_{ik} - wejściowy sygnał zewnętrzny,

x_{ik} - wyjście neuronu (i, k) ,

$$x_{ik} = 1/[1 + \exp(-u_{ik})]$$

Ogólnie funkcja energii sieci przedstawiona jest wzorem

$$E = -0.5 \mathbf{x}^T \mathbf{W} \mathbf{x} - \mathbf{R}^T \mathbf{x}$$

przy czym $\mathbf{x}$ - m -wymiarowy wektor, $m = n^2$,

$\mathbf{R}$ - m -wymiarowy wektor,

$\mathbf{W}$ - $m \times m$ wymiarowa macierz.

Funkcję energii:

$$E = E_1 + E_2 + E_3 + E_4$$

E_1 – reprezentuje zasadę: każda ramka posiada jedno i tylko jedno subźródło, E_2 –

wymaga, aby wyjście każdego neuronu dążyło do zera lub jedynki, E_3 – zapewnia

koherentność c-obszarów, E_4 – reprezentuje funkcję celu F

E_1 –

$$F = 0.5 \sum_{i,j} \sum_{k,l} d_{kl} e_{ij} x_{ik} x_{jl}$$

przy czym d_{kl} – odległość pomiędzy ramką k , a ramką l ,

e_{ij} – współczynnik proporcjonalny do mocy źródła zawierającego oba subźródła i oraz j .

Jeśli ramki i i j należą do różnych źródeł wówczas $e_{ij} = 0$.

Składniki funkcji energii wyglądają następująco

$$E_1 = A_1 \left[\sum_i \left(\sum_k x_{ik} - 1 \right)^2 + \sum_k \left(\sum_i x_{ik} - 1 \right)^2 \right]$$

$$E_2 = A_2 \sum_i \sum_k x_{ik} (1 - x_{ik})$$

$$E_3 = A_3 \sum_{i,j} \sum_{k,l} c_{kl} e_{ij} x_{ik} x_{jl}$$

$$E_4 = A_4 \sum_{i,j} \sum_{k,l} d_{kl} e_{ij} x_{ik} x_{jl}$$

$$A_1, A_2, A_3, A_4 \in \mathbf{R}$$

$c_{kl} = 0$ jeśli ramki k i l należą do koherentnego c-obszaru, w innym wypadku $c_{kl} = 1$.

Przedstawiona sieć neuronowa symulowana na mikrokomputerze została użyta do rozwiązania naszego zadania optymalizacji. Uściślijmy dane projektowe przykładowego układu. Wymiary podłoża $50 \times 80 \times 0.6 \text{ mm}^3$, chłodzenie - konwekcja swobodna ($\alpha = 15 \text{ W/m}^2\text{K}$), liczba ramek = 40. Przyjęto następujące wartości stałych we wzorach: $\tau = 10$, $A_1 = A_2 = 3$, $A_3 = 1$, $A_4 = 0.1$. Wartości początkowe:

$$(iv1) \quad u_{ik}(0) = -\log_e(n-1), \quad \text{dla } i, k = 1, \dots, n$$

$$(iv2) \quad u_{ik}(0) = 0$$

$$(iv3), (iv4) \quad u_{ik}(0) = \text{wartości losowe}$$

Pierwszy warunek (iv1) oznacza, że $x_{ik}(0) = 1/n$, wówczas składnik E_1 funkcji energii zeruje się w punkcie startowym. Warunek (iv2) ustala wartość każdego neuronu równą 0.5. Warunki (iv3) i (iv4) reprezentują losowo wybrane wartości.

Równanie (6.21) rozwiązano przy pomocy metody Rungego - Kutty czwartego rzędu ze zmiennym krokiem h . Jako wartość początkową h przyjęto 0.0001.

Dla różnych warunków początkowych (iv1-iv4) otrzymano cztery rozwiązania. Rozwiązania różnią się zarówno kształtem c-obszarów jak i liczbą iteracji *ITER*, potrzebnych do osiągnięcia lokalnego minimum funkcji energii. Wyniki przedstawiono na rysunku i w tablicy 1. Dla każdej tak otrzymanej topografii obliczono rozkład temperatury na podłożu metodą analityczną.

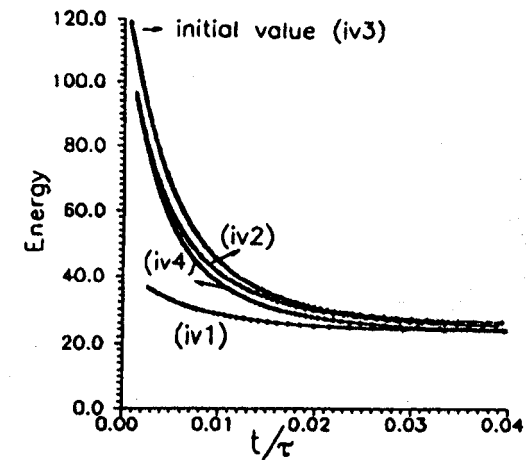
Przy pomocy wzoru

$$D = \sqrt[2N]{\frac{1}{N} \sum_{i=1}^N (T_i - T_m)^{2N}}$$

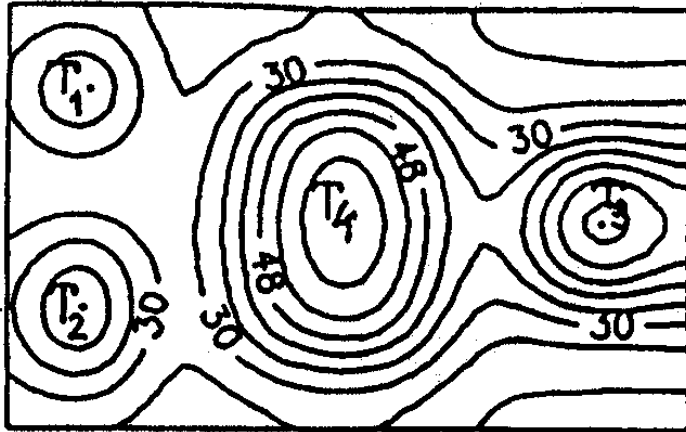
(przy czym T_i - temperatura obliczona w środku i -tego źródła, T_m - temperatura średnia układu, N - liczba źródeł ciepła) porównano poszczególne rozwiązania (tab.1). Analizując wyniki należy stwierdzić, że jakkolwiek otrzymano różne rozwiązania (różne minima lokalne funkcji energii), to z praktycznego punktu widzenia są one równoważne. Szczególną uwagę zwraca przypadek (iv1), ze względu na minimalną liczbę iteracji.

Tablica .1 Różne możliwości podziału podłoża na c-obszary

Rys.	S1	S2	S3	S4	S	D °C	Iv	ITE R
a	8	12	14	18	52	26.2	4	146
b	8	10	14	18	50	27.5	1	46
c	8	10	16	20	54			
d	10	12	14	22	58	25.3	2	172
e	10	14	14	18	56			
f	10	14	14	18	56			
g	8	10	22	18	58			
h	8	10	14	22	54	27.7	3	190

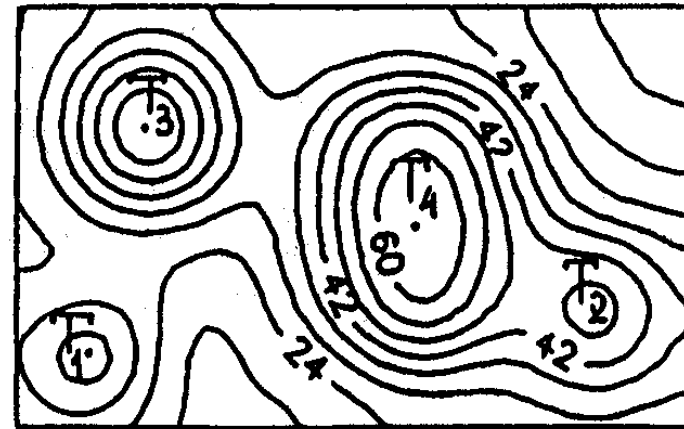


$T_1=40.5, T_2=46.2, T_3=61.5, T_4=64.5.$



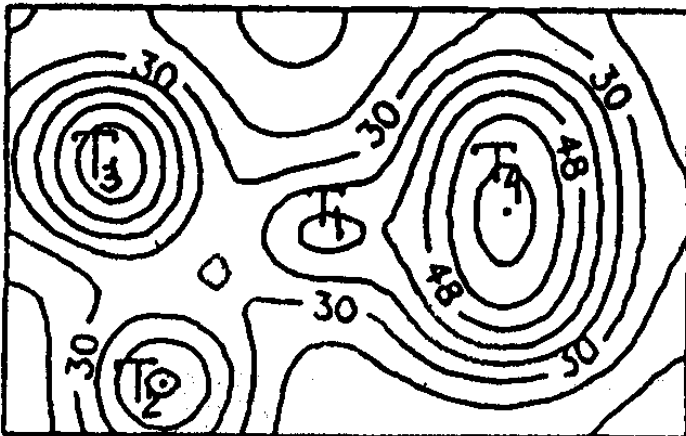
(iv1)

$T_1=38.4, T_2=49.3, T_3=59.1, T_4=65.6.$



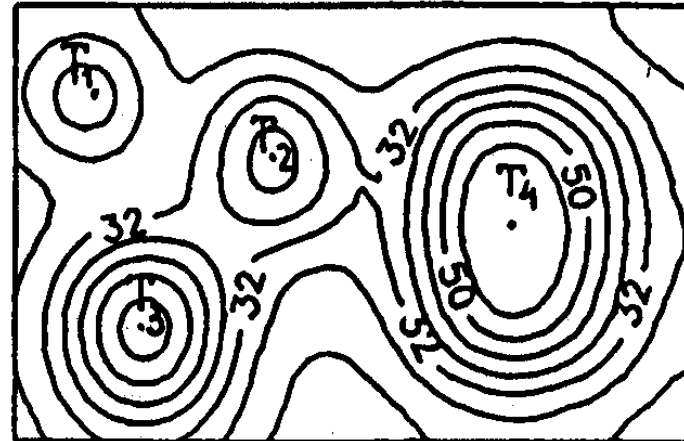
(iv3)

$T_1=43.2, T_2=49.2, T_3=58.5, T_4=62.4.$



(iv2)

$T_1=41.6, T_2=46.1, T_3=58.8, T_4=63.6.$



(iv4)

Różne rozwiązania zadania optymalizacji

Ze wzrostem liczby źródeł ciepła wzrasta szybko liczba neuronów w sieci Hopfielda. Dla uproszczenia proponuje się wprowadzić hierarchiczny podział podłoża. Każdy poziom hierarchii nie powinien przekraczać czterech części. Wstępnie źródła ciepła grupowane są w abstrakcyjne źródła o mocy sumarycznej. Otrzymujemy nowe, uproszczone zadanie o mniejszej liczbie źródeł. Po jego rozwiązaniu przechodzimy do poziomu coraz to niższego, aż osiągniemy poziom najniższy, w którym poszukujemy rozkładu rzeczywistych źródeł.

Dla naszego zadania optymalizacji można dokonać przykładowego zgrupowania: $P_1 + P_2 = 1.0\text{W}$, $P_3 = 1.0\text{W}$, $P_4 = 2\text{W}$. Podłoże podzielone jest na trzy c-obszary. W niższym poziomie c-obszar "1+2" dzieli się na dwa oryginalne c- obszary.

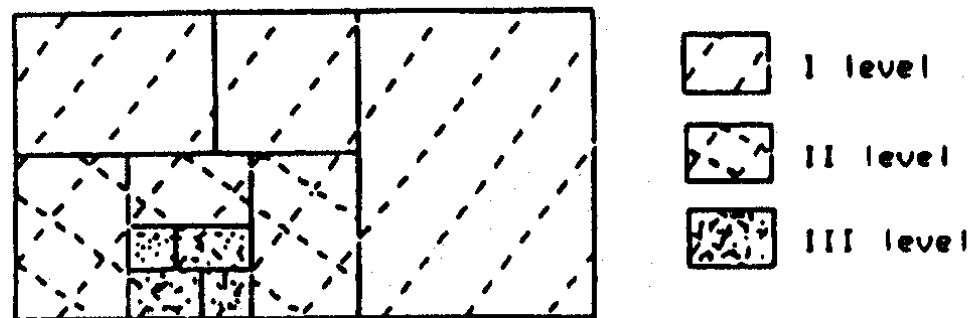
Dzięki dwupoziomowemu hierarchicznemu podziałowi osiągnięto znaczne skrócenie czasu obliczeń komputerowych. Pierwszy poziom wymaga sieci z zaledwie 16 neuronami. Problem drugiego poziomu (niższego) rozwiązuje sieć składającą się ze 100 neuronów. Czas obliczeń jest ponad 100 razy krótszy w porównaniu z siecią niehierarchiczną. Wyniki przedstawiono w tablicy 2. Wartości wskaźnika D są tylko nieznacznie gorsze.

Tablica 2 Podział hierarchiczny podłoża

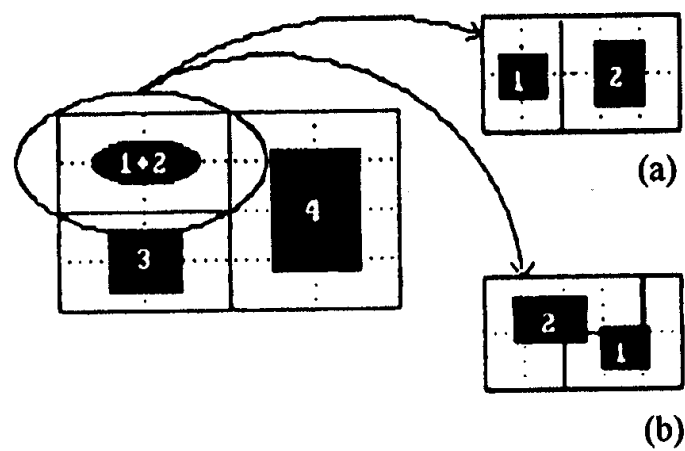
Rys.	$S1$	$S2$	$S\ 1,2$	D °C	iv	$ITER$
a	12	14	26	28.0	1	42
b	14	16	30	29.2	3	98

Taki sam podział hierarchiczny zastosowano do zadania, w którym opisany mikroukład z podłożem alundowym przyklejono centrycznie do aluminiowego wymiennika ciepła o wymiarach $100 \times 200 \times 2\text{mm}^3$.

W tym wypadku estymator D dla przypadków (a) i (b) - tab. 2 wynosił odpowiednio 16.2°C i 16.8°C .



Hierarchia trójpoziomowa



Hierarchiczny rozkład czterech źródeł ciepła

