Sieci neuronowe w optymalizacji

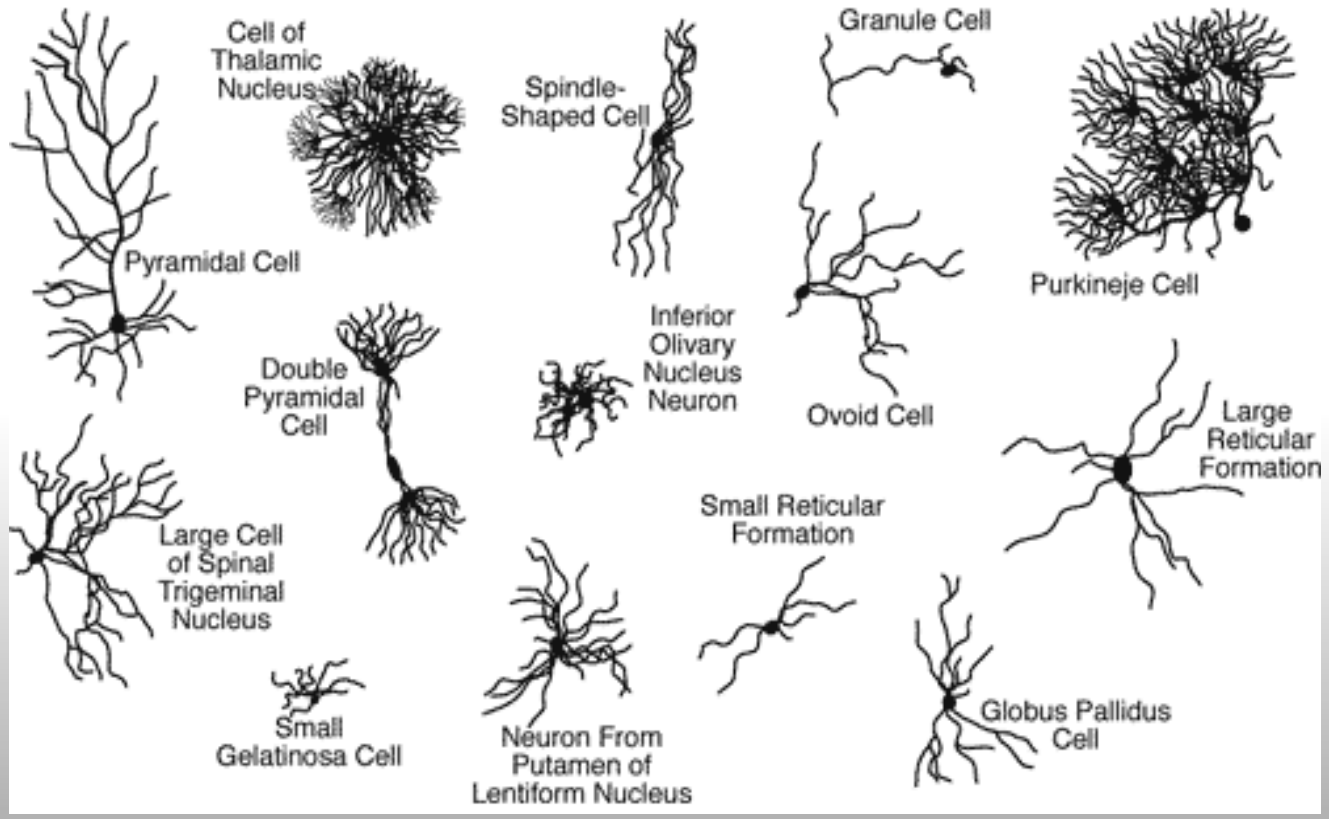
Bartłomiej Gałkowski

Dlaczego chcielibyśmy innego podejścia?

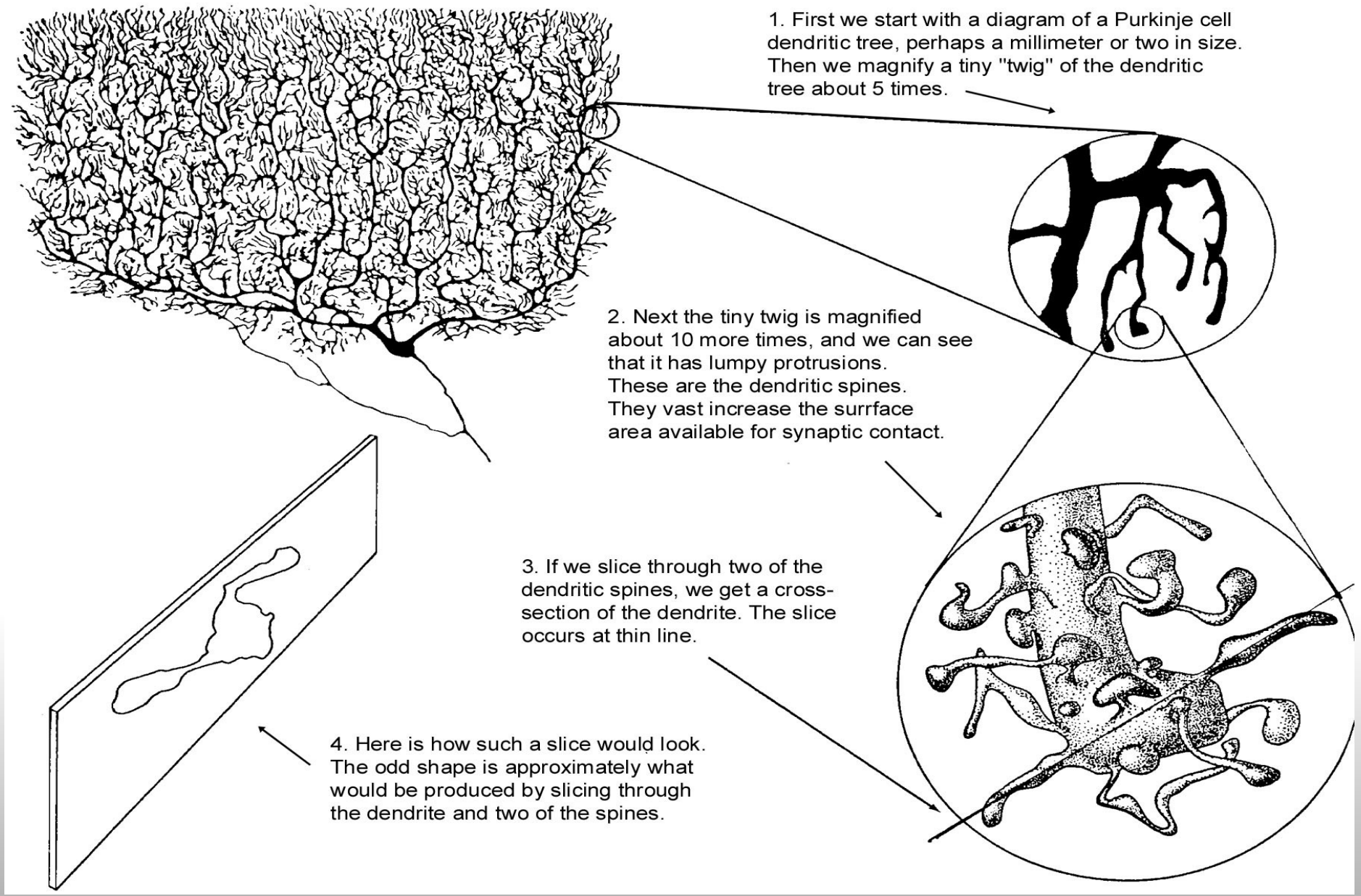
Wady i zalety tradycyjnych systemów obliczeniowych

Łatwe	Trudne
Szybka arytmetyka	Interakcja z rzeczywistymi zaburzonymi danymi i adaptacja
Wykonywanie algorytmów w dokładnie taki sposób jaki zostały zapisane	Obliczenia masywnie równoległe
	Odporność na awarie

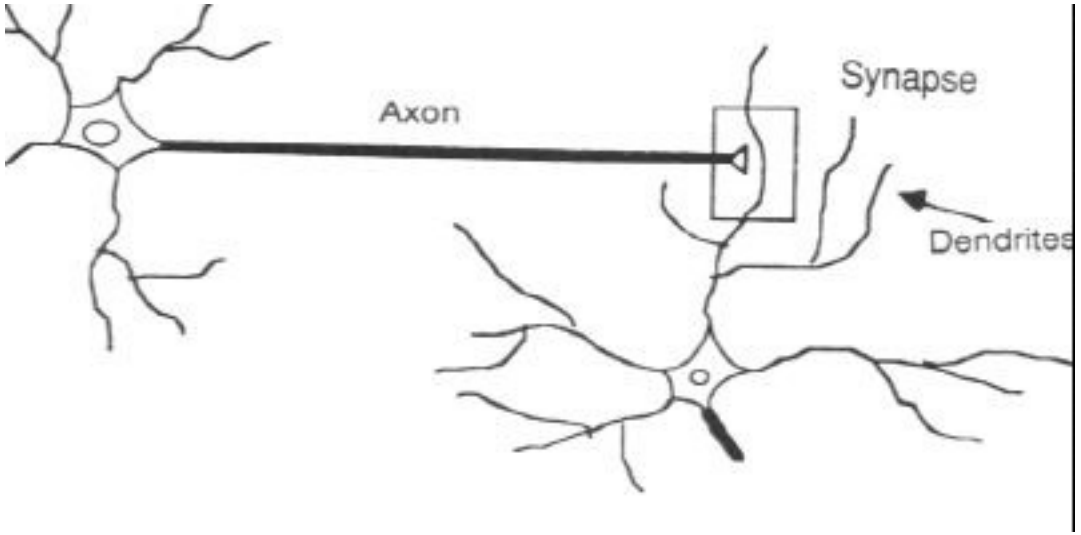
Różne rodzaje neuronów

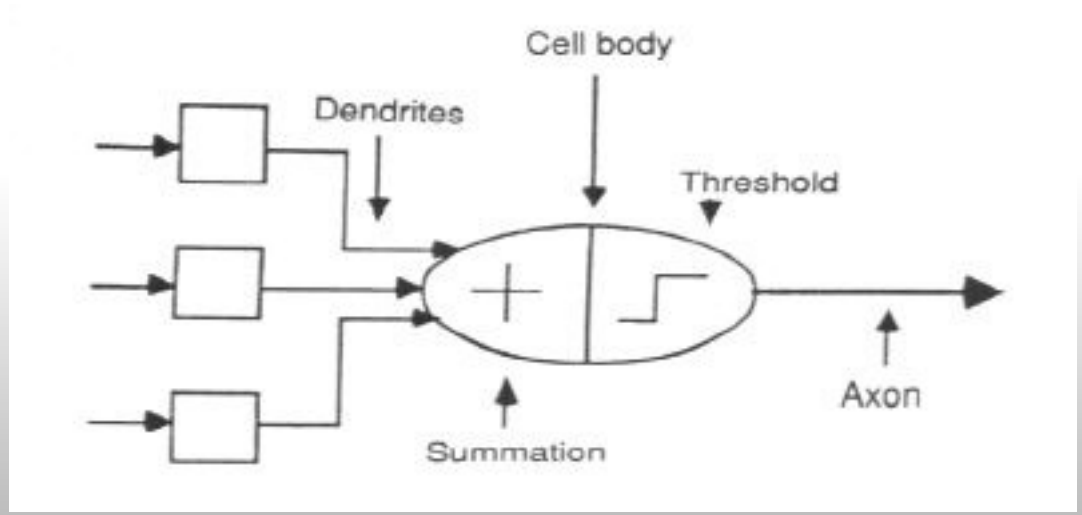


Inspiracja: jak wygląda prawdziwy neuron

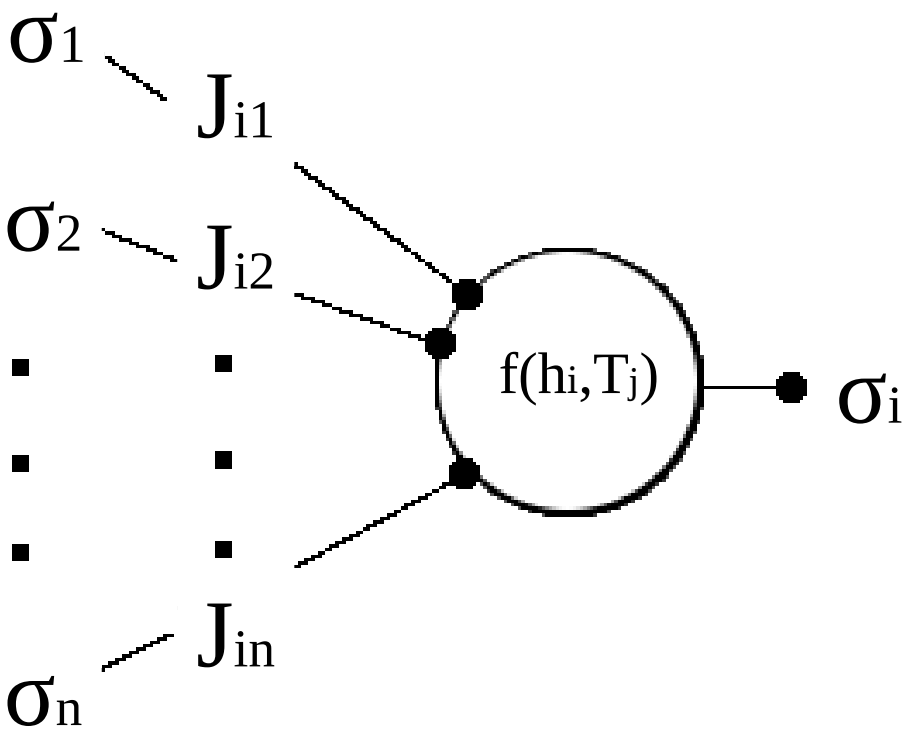


Jak można je opisać





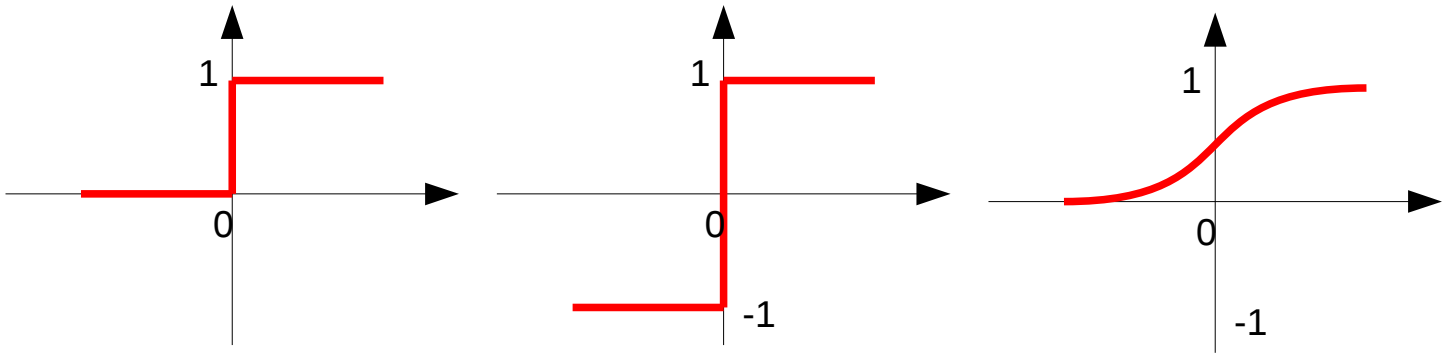
Neuron McCullocha-Pittsa



$$\sigma_i(t) = f\left(h_i(t) - T_i\right)$$

$$h_i(t) = \sum_{j=1}^{n} J_{ij}\sigma_j(t-1)$$

Funkcje aktywacji neuronu



Heavside'a Θ

Funkcja signum

f(x)=1/(1+exp(-gx))

Inne opisy neuronów

- Neurony pamiętające swój poprzedni stan:

$$S_i(t+1) = f\left[\sum_{j=1}^{N} J_{ij} S_j - \sum_{r=0}^{t} k^r S_i(t-r) - T_i\right]$$

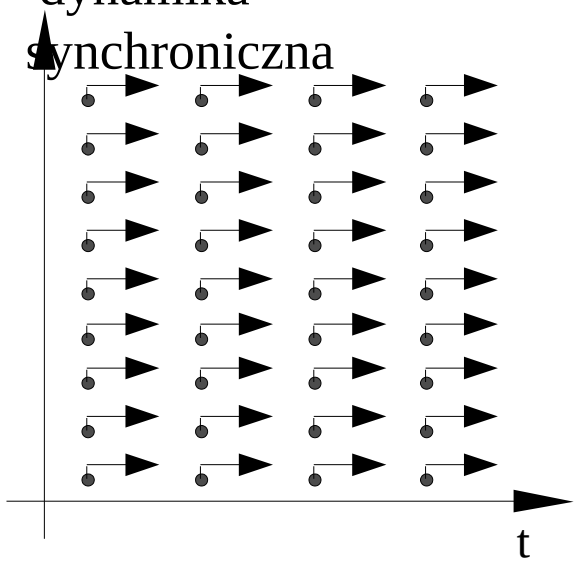
- Neurony analogowe:

$$C_i \frac{du_i}{dt} = \sum_{j \neq i}^{N} J_{ij}\, f(ui) - u_i / R_i + I_i$$

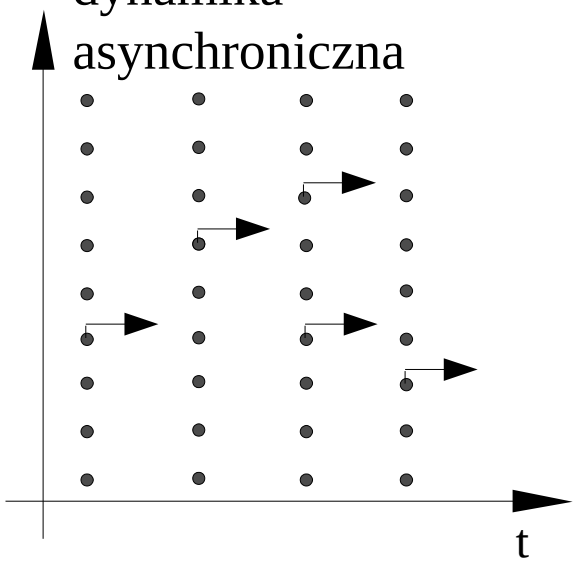
$$V_i = V_0\, f(u_i)$$

Rodzaje dynamiki sztucznych sieci

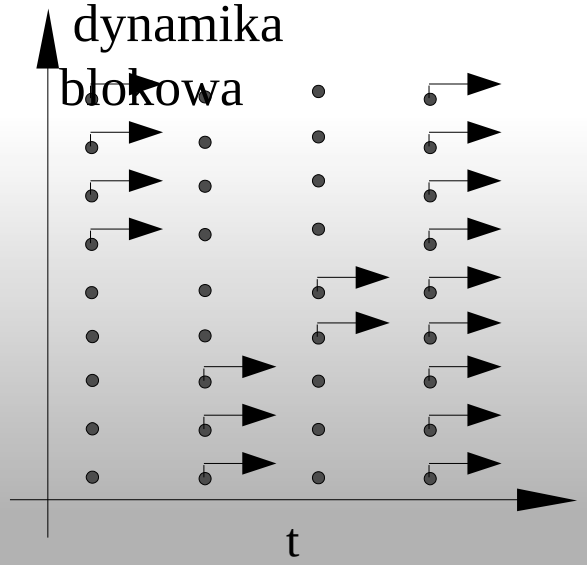
dynamika synchroniczna



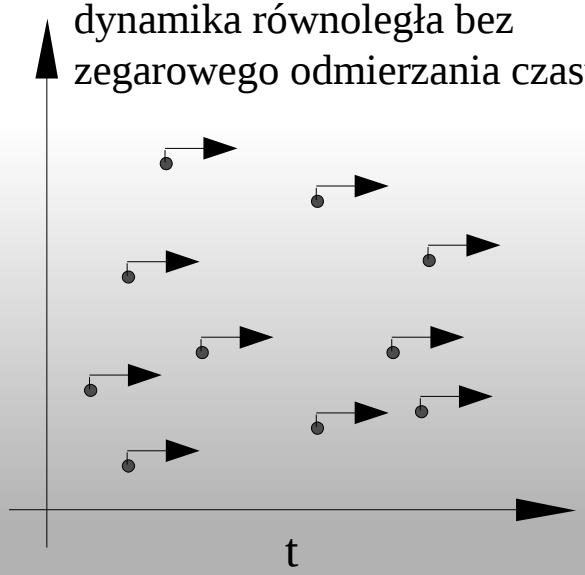
dynamika asynchroniczna



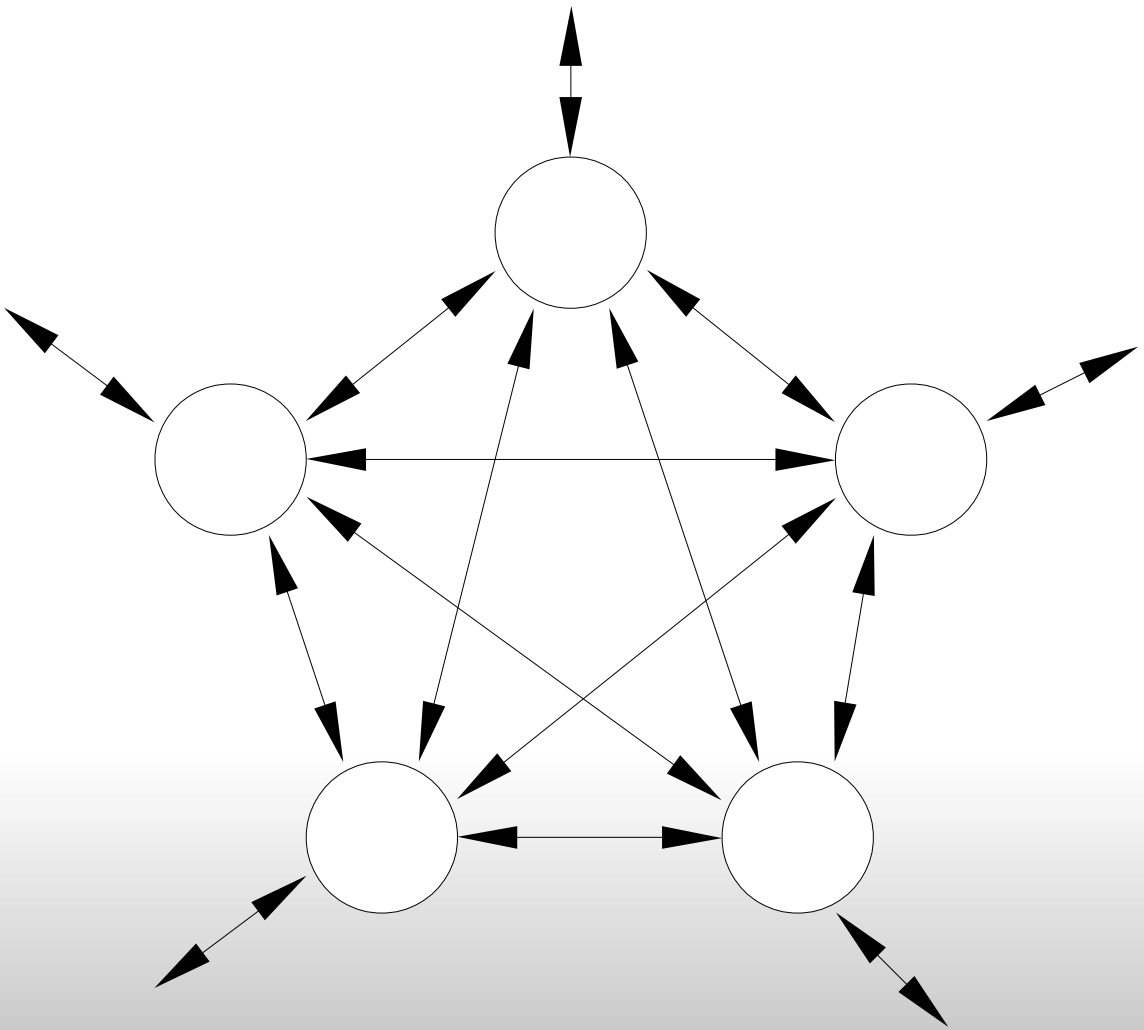
dynamika blokowa



dynamika równoległa bez zegarowego odmierzania czasu



Sieć Hopfielda



Sieć Hopfielda w rozpoznawaniu

- Zwykle neurony są tu dwustanowe
- Funkcja aktywacji Heavsidea
- Reguła Hebba używana do nauczenia sieci:

$$\Delta J_{ij} = \frac{1}{N} \xi_i \xi_j$$

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i \xi_j$$

N – liczba neuronów
P – liczba wzorców
J – macierz wag

Jak przebiega przykładowe odpytanie sieci Hopfielda

Macierz J połączeń między neuronami ma postać:

	Neuron 1	Neuron 2	Neuron 3	Neuron 4
Neuron 1	0	-1	1	-1
Neuron 2	-1	0	-1	1
Neuron 3	1	-1	0	-1
Neuron 4	-1	1	-1	0

Jeśli przedstawimy takiej sieci wzorzec:
$\xi = 0101$

$N1 = \Theta(-1 + -1) = 0$
$N2 = \Theta(0 + 1) = 1$
$N3 = \Theta(-1 + -1) = 0$
$N4 = \Theta(1 + 0) = 1$

to mówimy że rozpoznaje jeśli ustabilizuje się w stanie będącym wzorcem

Otrzymywanie macierzy J

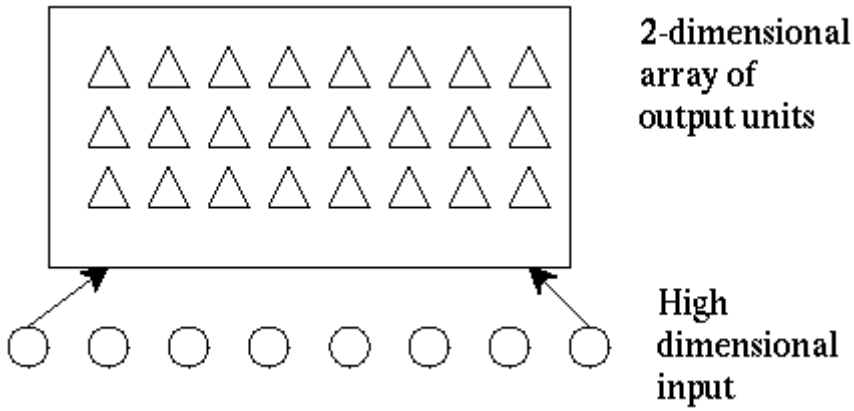
- Nasz wzorzec ξ = [0,1,0,1] przedstawiamy w postaci binormalnej ξ = [-1,1,-1,1] i wykorzystujemy wspomniany wzór:

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i \xi_j$$

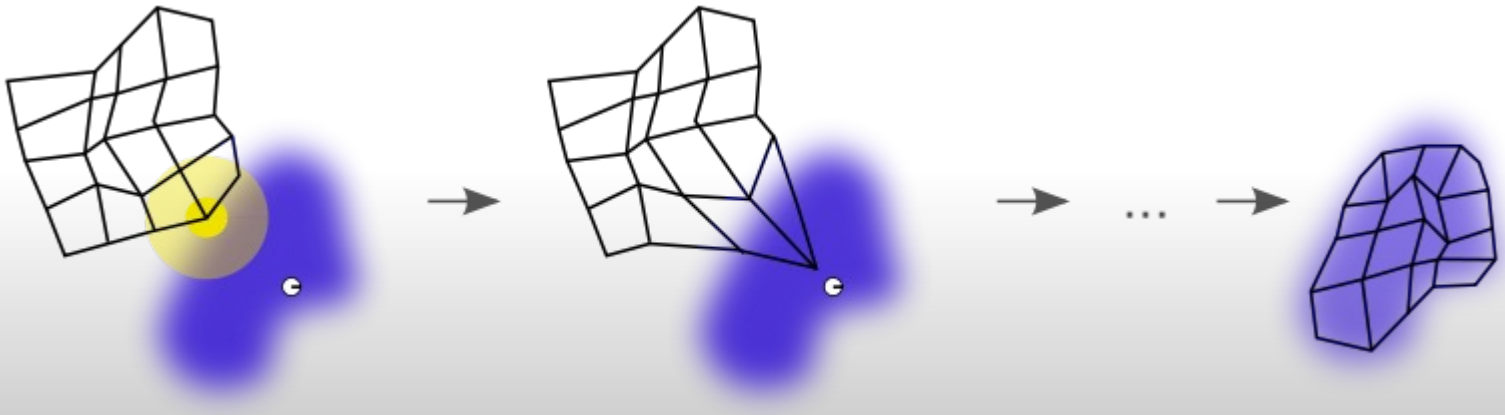
Jak użyć sieci Hopfielda do TSP

- Możemy stworzyć sieć o N^2 neuronach
- Neuron N_{ia} jest wzbudzony tylko wtedy gdy i-te miasto jest na a-tym miejscu w cyklu
- Odległości podaje się jako wagi połączeń między neuronami: $d_{ij} = J_{ijab}$

Pomysł na sieć Kohonena



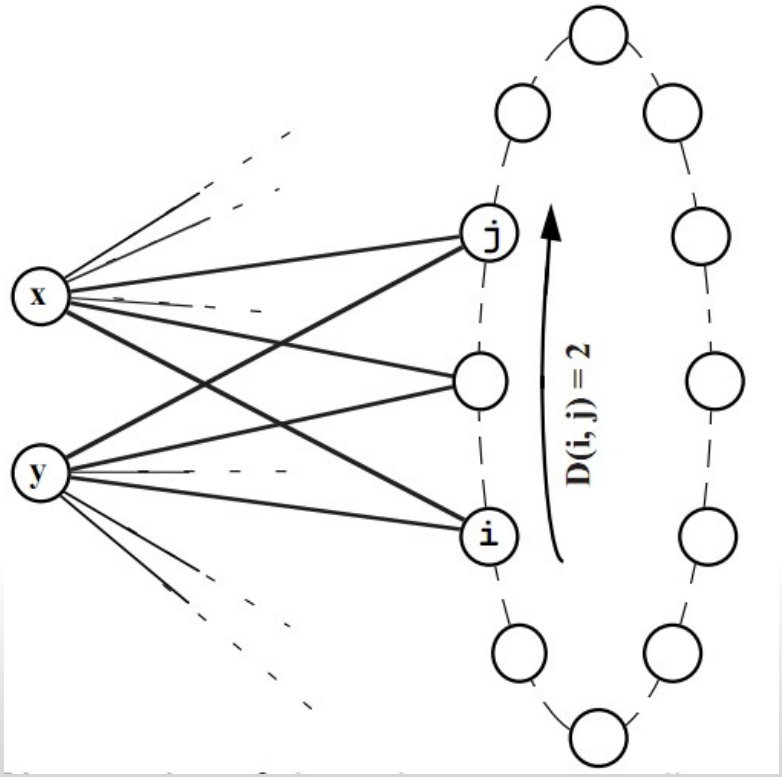
Stworzyć dwie warstwy neuronów:
Wejściową (otrzymująca wszystkie wymiary wejścia)
zachowującą topologię rozwiązania



Rozwiązywanie TSP za pomocą sieci Kohonena

- W naszym problemie komiwojażera n miast ma losowe współrzędne z przedziału (0,1)
- Sieć ma dwa neurony wejściowe dla każdej współrzędnej miasta
- Neurony wyjściowe tworzą pierścień i każdy ma dwie wartośći (w_x,w_y) w wyniku podania współrzędnych zwraca o = (w_x,w_y) • (x,y)

Rozwiązywanie TSP za pomocą sieci Kohonena



Algorytm

1. Ustaw wagi neuronów na losowe wartości (0,1)
2. Wybierz losowe miasto ξ i podaj jego współrzędne do neuronów wejściowych
3. Znajdź neuron wyjściowy który zwraca największą wartość poddaj jego i jego sąsiadów treningowi:

$$w_i = w_i + \alpha(\xi - w_i) \quad \forall i : D(i, m) < d$$

4. Uaktualni parametry d oraz α
5. Wróć do kroku 2

Usprawnienia

- Początkowe ustawienie neuronów wyjściowych tworzy okrąg
- Wartość d i α maleje wykładniczo z współczynnikiem p = 0.995
- Neuronów jest dwa razy więcej niż miast

Bibliografia

- Robert A. Kosiński *Sztuczne siecie neuronowe: dynamik nielinowa i chaos* 2002 WNT
- Jeff Heaton *Introduction to Neural Networks with Java* 2000 Heaton Research, Inc
- Marco Budinich *A Self-Organising Neural Network for the Traveling Salesman Problem that is Competitive with Simulated Annealing Neural Computation Volume 8, Issue 2*