

# MODELOWANIE RZECZYWISTOŚCI

**Daniel Wójcik**

Instytut Biologii Doświadczalnej PAN  
Szkoła Wyższa Psychologii Społecznej

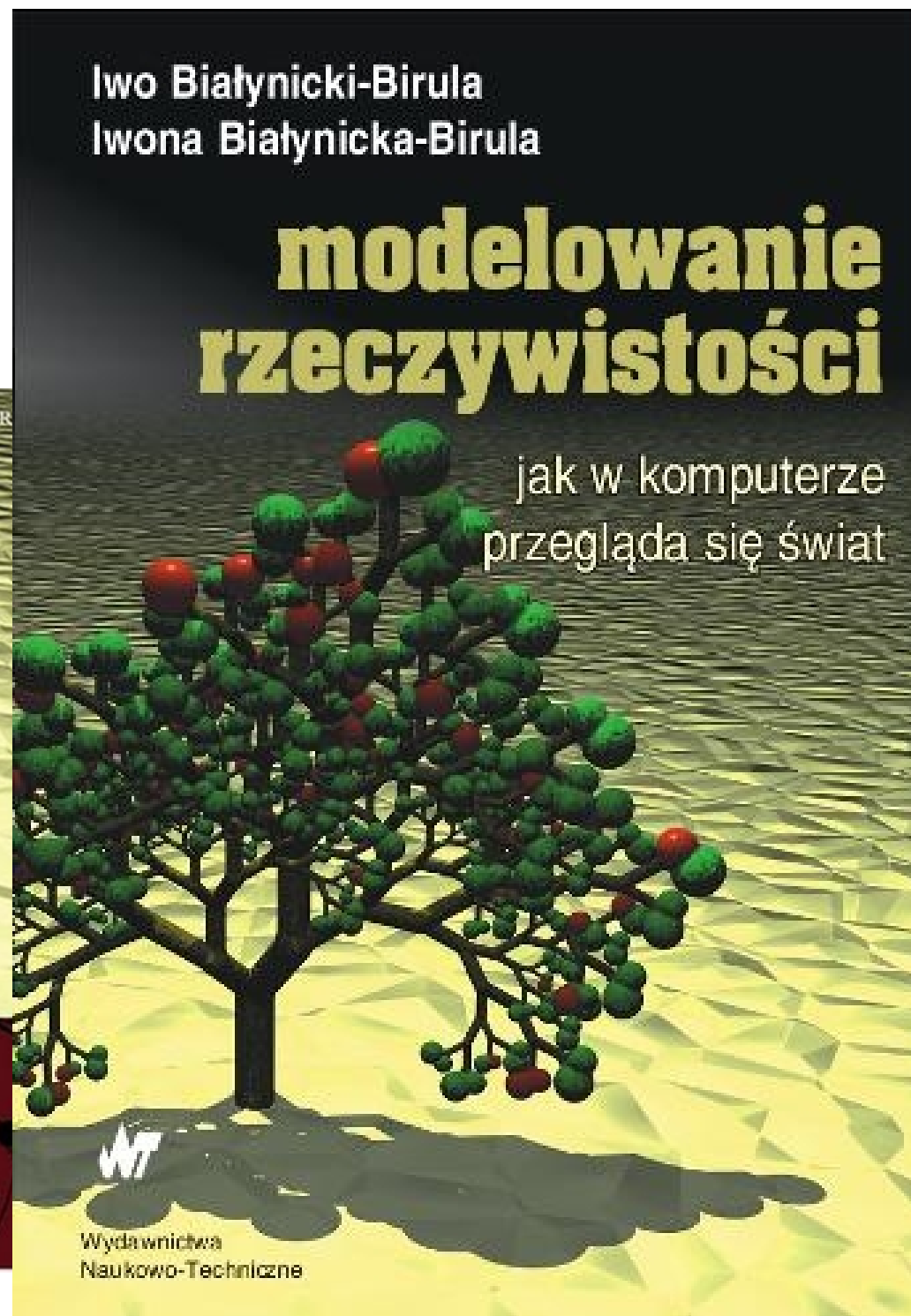
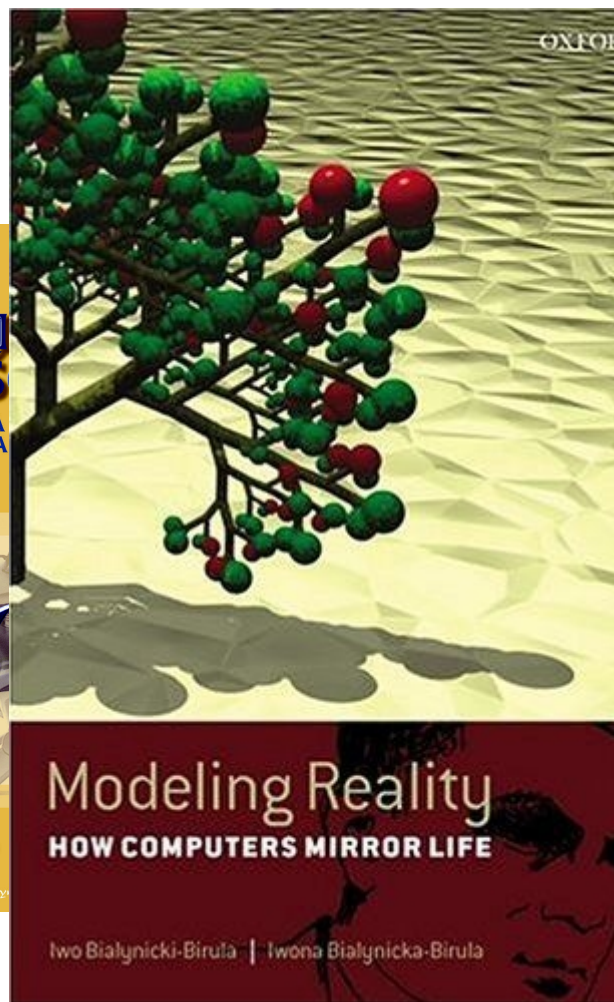
[d.wojcik@nencki.gov.pl](mailto:d.wojcik@nencki.gov.pl)  
[dwojcik@swps.edu.pl](mailto:dwojcik@swps.edu.pl)

tel. 022 5892 424

<http://www.neuroinf.pl/Members/danek/swps/>

# Podręcznik

Iwo Białynicki-Birula  
Iwona Białynicka-Birula



# Komputer jako model mózgu

- Komputer jest modelem mózgu w tym sensie, że również wykonuje obliczenia
- **Model von Neumanna:**
  - Procesor centralny (CPU) – „myślenie”
  - Pamięć – „zapamiętywanie”
- Działanie komputera:
  - Pobierz z pamięci instrukcję
  - Pobierz z pamięci dane do wykonania instrukcji
  - Wykonaj instrukcję
  - Zapisz uzyskane dane w pamięci

# Wady komputera von Neumanna

- Brak odporności na uszkodzenie elementów
- Sekwencyjne działanie
- Brak adaptacji
  
- **Alternatywa?**

# Sztuczne sieci neuronowe

**Sieci neuronowe** to modele obliczeniowe, których funkcjonalność i struktura wzorowana jest na mózgu.

Charakteryzuje je:

- Rozproszone i równoległe przetwarzanie informacji
- Odporność na uszkodzenia części elementów
- Możliwość uczenia

# „Mózg”

- Mózg składa się z komórek nerwowych – neuronów
- Z ciała komórki neuronu odchodzą dendryty, które razem z ciałem komórki zbierają informację przychodzącą
- Jeżeli sumaryczne pobudzenie komórki przekroczy próg aktywacji, komórka generuje impuls wysyłany aksonem do sąsiadów

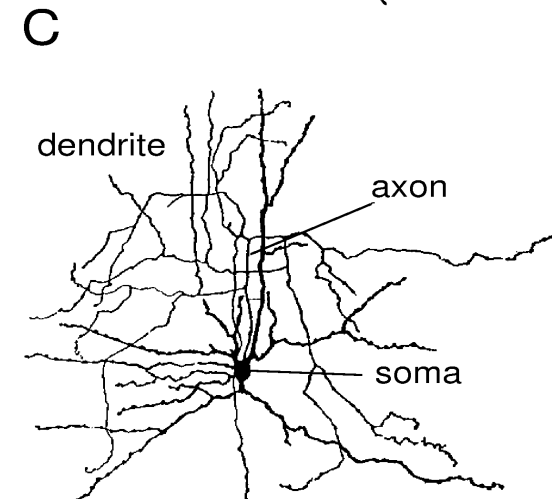
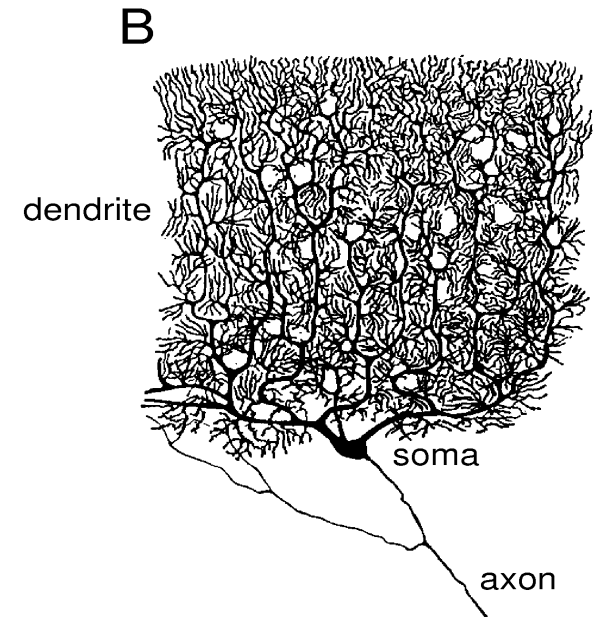
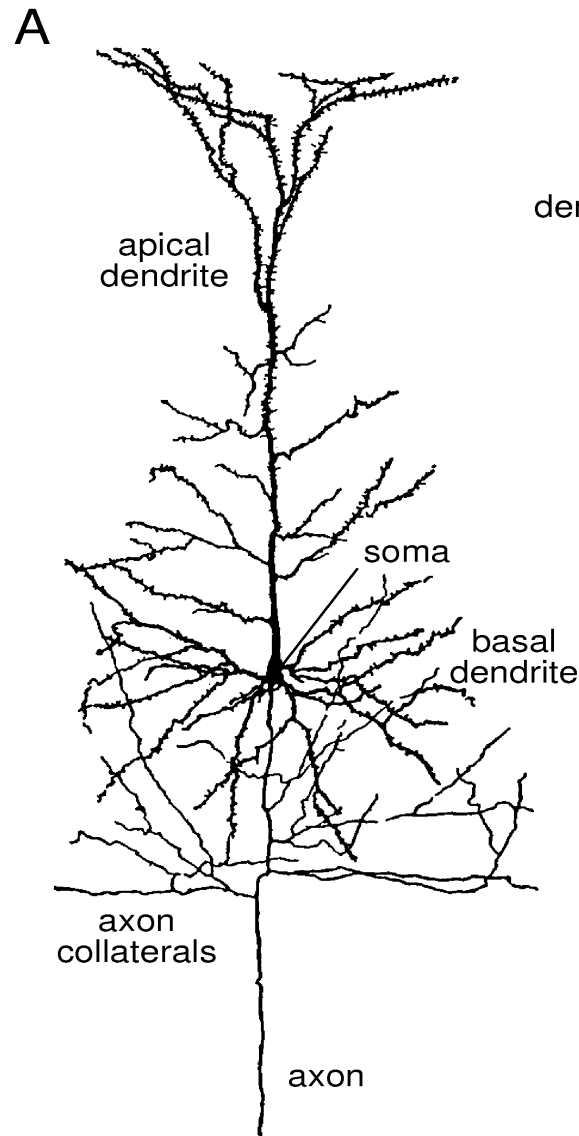
# Komórka nerwowa

- Typowe neurony

A. Komórka piramidalna korowa

B. Komórka mózdzka

C. Gwiazdzista komórka kory mózgowej

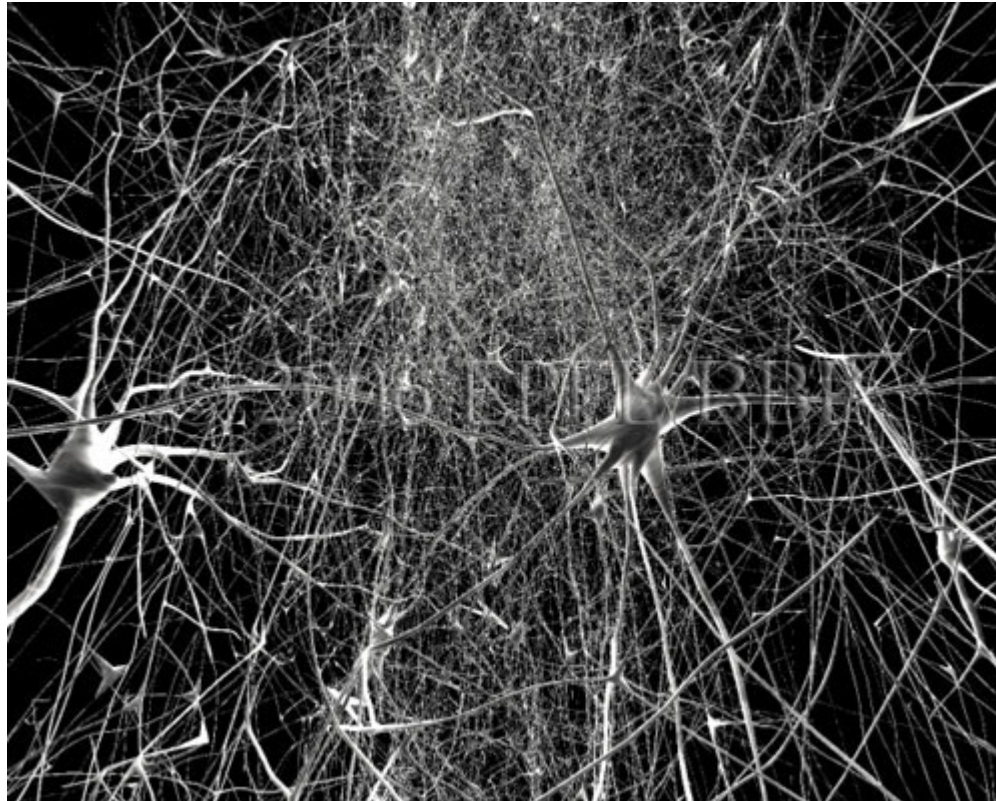


# Komórka nerwowa





# Połączenia w układzie nerwowym



# Połączenia w układzie nerwowym



# Synapsa

- Połączenie między neuronami nazywamy synapsą
- Kiedy sygnał dochodzi do synapsy, wysyłana jest substancja chemiczna (neurotransmitter), która aktywuje komórkę postsynaptyczną
- Komórka presynaptyczna może zwiększać lub zmniejszać aktywność komórki postsynaptycznej
- Waga synapsy może się zmieniać podczas przetwarzania informacji

# Teoria Hebba (1949)

- Zmiany wag synaps leżą u podłoża procesów zapamiętywania i uczenia się

# Szybkość przetwarzania informacji

- Neurony działają w skali milisekund, czyli w przybliżeniu wykonują ok. 1000 operacji na sekundę
- Procesory w komputerach są około milion razy szybsze
- Ale w mózgu naraz działa  $10^{11}$  neuronów – przetwarzanie równoległe

# Sztuczna sieć neuronowa

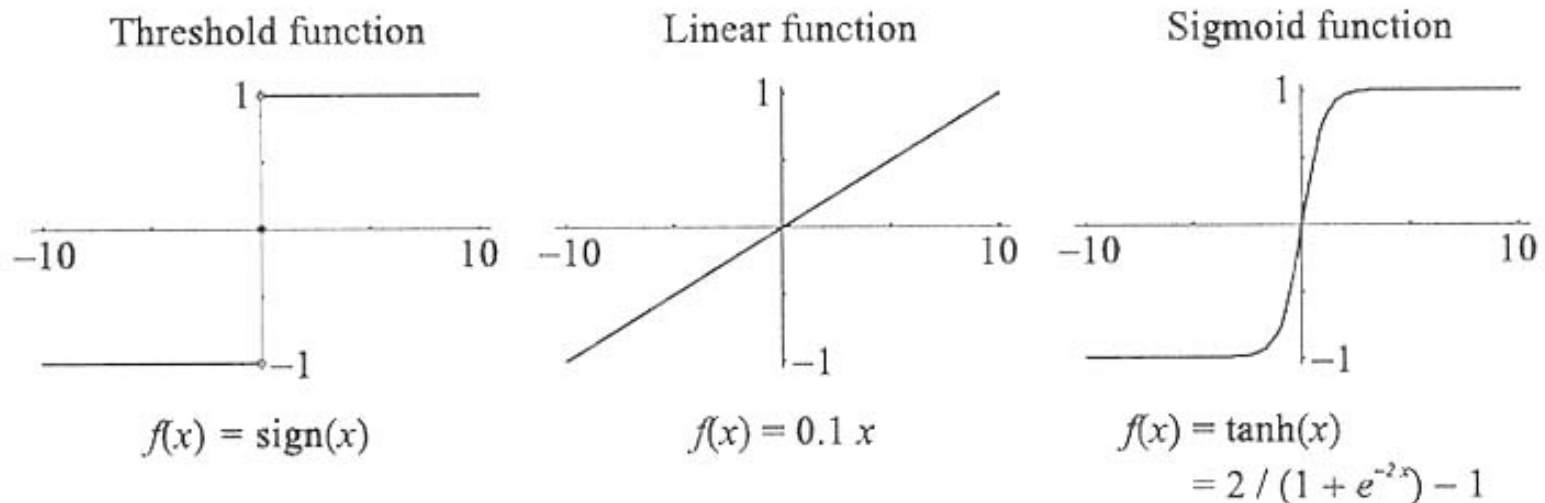
- Powyższe inspiracje biologiczne były inspiracją do stworzenia sztucznych sieci neuronowych – uproszczonych modeli mózgu przetwarzających informację w sposób rozproszony i równoległy.
- Realizuje się je albo programowo albo sprzętowo (układy scalone o dużej skali integracji)

# Sztuczny neuron

- Podstawowym elementem sztucznych sieci neuronowych jest schematyczny model neuronu
- Jego stan odpowiada częstotliwości impulsów generowanych przez komórkę
- Sieci neuronowe składają się z połączonych sztucznych neuronów

# Sztuczny neuron

- Połączenia odpowiadają synapsom, z każdym jest skojarzona pewna waga. Kiedy sygnał przechodzi przez połączenie, mnożony jest przez wagę
- Sztuczny neuron na wyjściu podaje ważoną sumę przychodzących sygnałów, którą czasem dodatkowo się moduluje (**funkcją aktywacji**)





# Sieci neuronowe

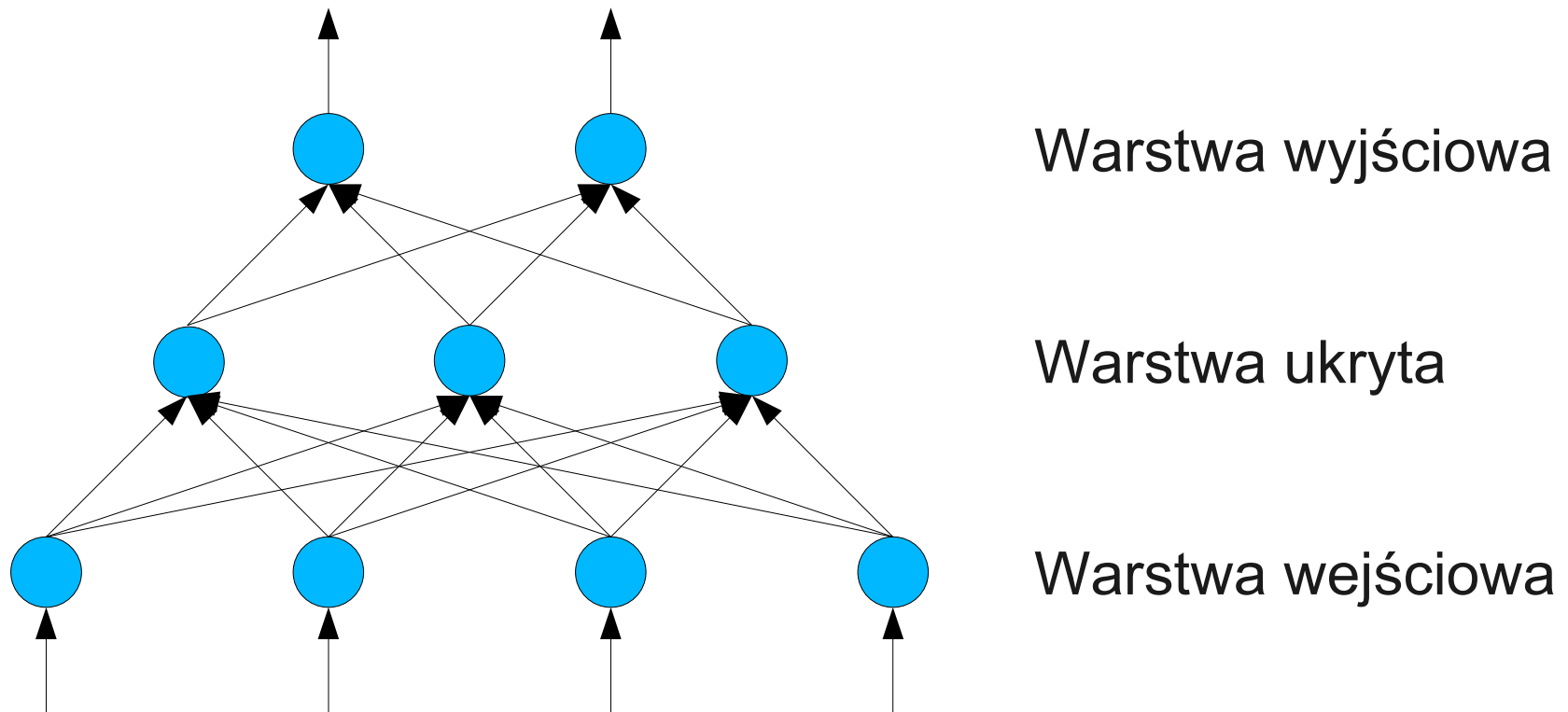
Kiedy chcemy rozwiązać dany problem przy pomocy sieci neuronowej napotykamy dwa główne problemy:

- Jak zaprojektować sieć (znaleźć graf skierowany reprezentujący sieć)
- Jak nauczyć sieć (dobrać wagi by sieć wykonywała żądane zadanie)

Nie wiemy jak natura rozwiązuje te problemy (ewolucja), nie mamy też ogólnej metody konstrukcji sztucznych sieci neuronowych.

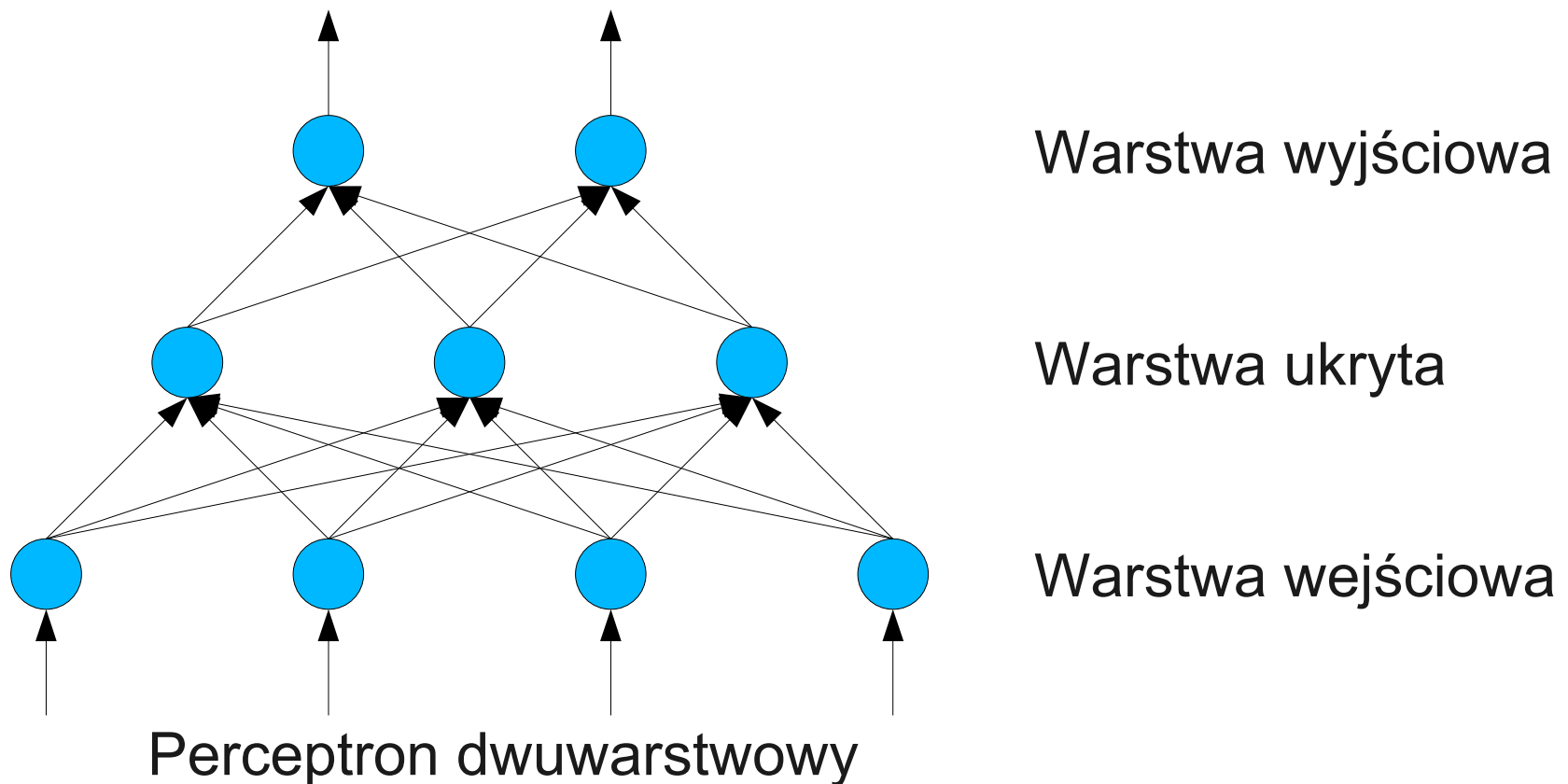
# Sieci neuronowe

Sieci neuronowe jednokierunkowe (**feedforward neural networks**) są budowane warstwowo. Składają się z **warstwy wejściowej**, która odpowiada neuronom sensorycznym, i **wyjściowej**, która odpowiada neuronom motorycznym. Często także z kilku tzw. **warstw ukrytych**, które odpowiadają interneuronom układu nerwowego.



# Perceptrony

Najprostszą i najpopularniejszą siecią neuronową jest **perceptron wielowarstwowy**. Jest siecią jednokierunkową (**feedforward**), to znaczy nie zawierającą cykli. Informacja podana jej na wejściu przechodzi taką sieć raz i stabilizuje się na wyjściu.



# Jak działa perceptron?

- Oznaczmy węzły 1, 2, ..., i, ... warstwy k-tej przez  $N_i^k$
- Ustawiamy stan  $S_i^1$  każdego węzła  $N_i^1$  warstwy wejściowej na i-tą składową sygnału wejściowego  $I_i$
- W kolejnych warstwach od drugiej do wyjściowej ( $k=M$ ) postępujemy następująco:

# Jak działa perceptron?

- W każdym węźle  $N_i^k$  w warstwie  $k$  policz ważoną sumę wchodzących sygnałów, zastosuj funkcję aktywacji  $f$ ; wynik przypisz stanowi  $S_i^k$  tego węzła. Inaczej mówiąc

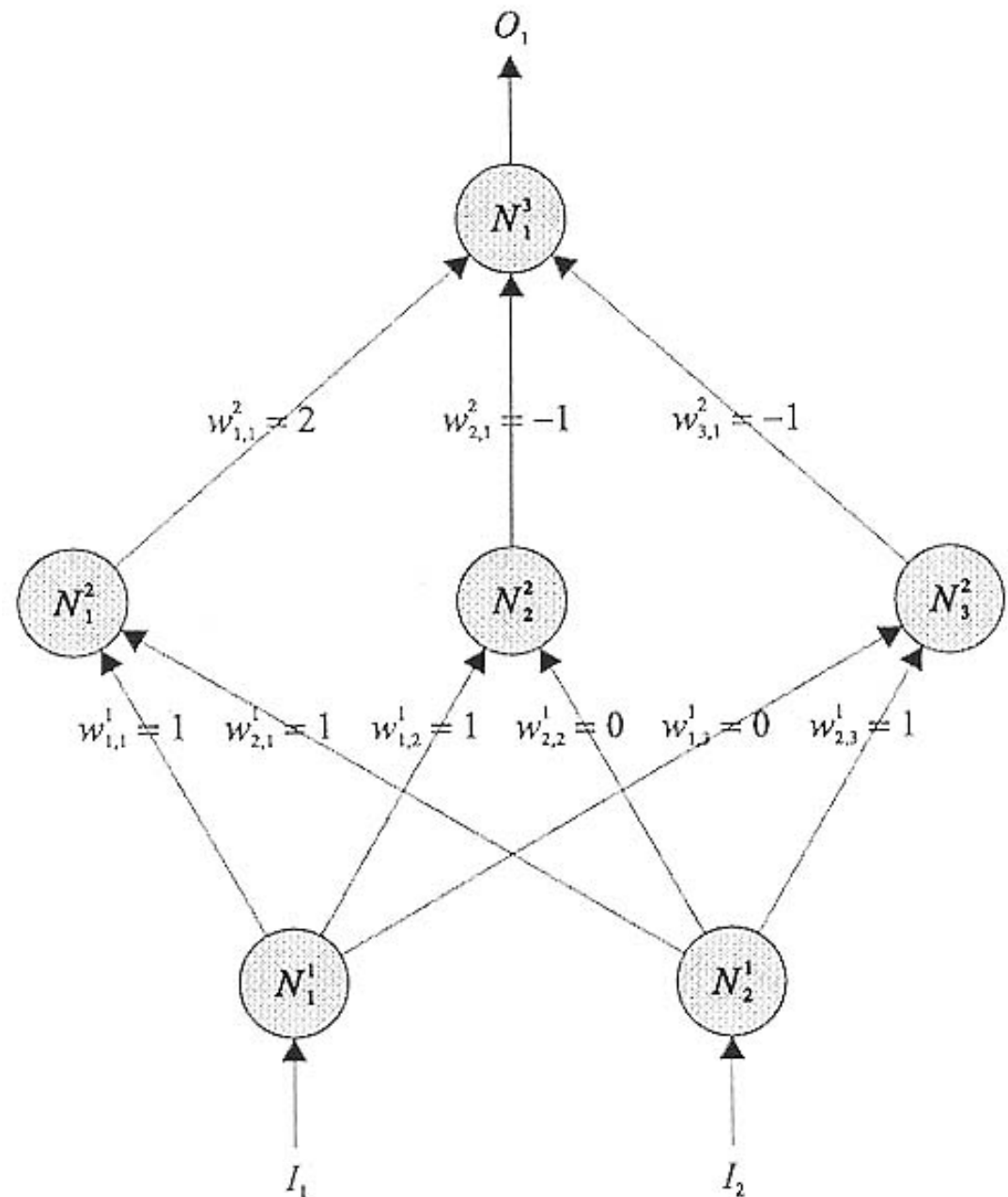
$$S_i^k = f \left( \sum_{j=1}^{L_{k-1}} w_{j,i}^{k-1} S_j^{k-1} \right)$$

gdzie  $L_{k-1}$  to liczba węzłów w warstwie  $k-1$ ,  $w_{j,i}^{k-1}$  jest wagą połączenia od węzła  $N_j^{k-1}$  do węzła  $N_i^k$

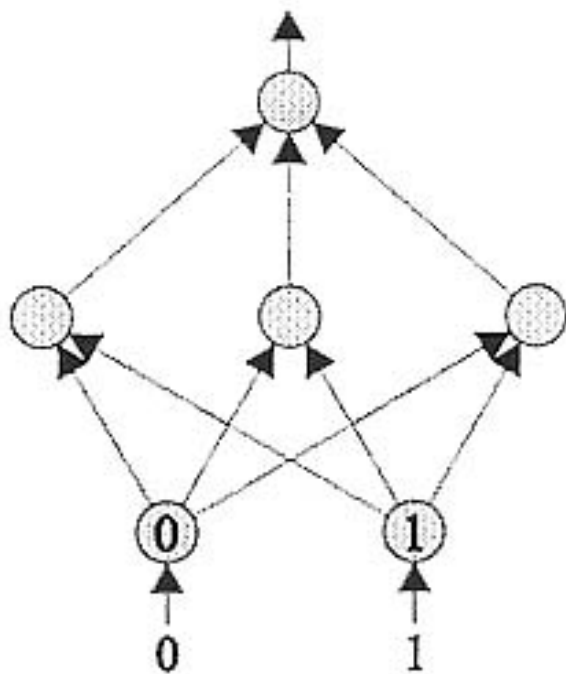
- Skopiuj stan  $S_i^M$  każdego węzła  $N_i^M$  na  $O_i$  –  $i$ -tą składową sygnału wyjściowego.

# Przykład: reguła XOR

- W roku 1969 M. Minsky i S. Papert udowodnili, że żaden perceptron jednowarstwowy nie może wyliczyć funkcji XOR.
- Okazało się, że może to zrobić perceptron dwuwarstwowy.

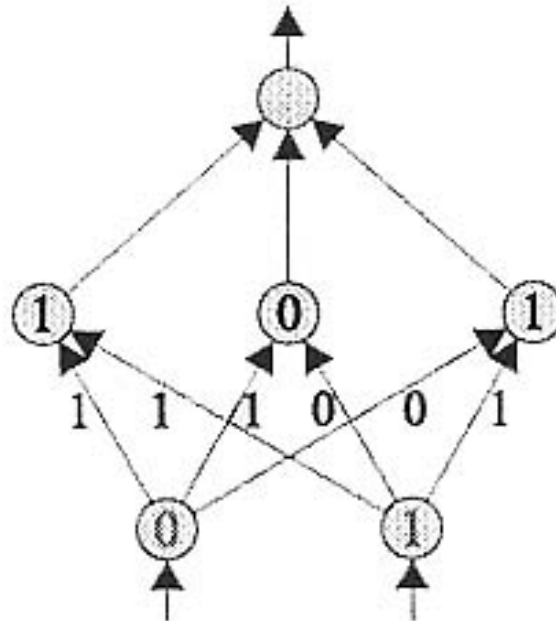


# Perceptron XOR w akcji



$$S_1^1 = I_1 = 0$$

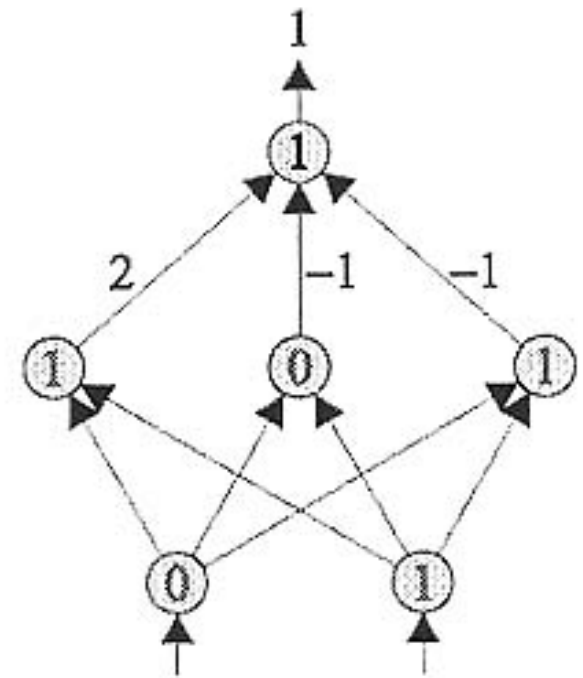
$$S_2^1 = I_2 = 1$$



$$S_1^2 = \text{sign}(1 \times 0 + 1 \times 1) = 1$$

$$S_2^2 = \text{sign}(1 \times 0 + 0 \times 1) = 0$$

$$S_3^2 = \text{sign}(0 \times 0 + 1 \times 1) = 1$$



$$O_1 = S_1^3 = \text{sign}(2 \times 1 - 1 \times 0 - 1 \times 1) = 1$$

# Uczenie sieci

- Uczenie sieci polega na zmianie wag połączeń tak, by funkcjonowała w pożądanym sposób.
- Zwykle – automatyczne dopasowywanie wag zgodnie z pewnym algorytmem uczenia.
- Kiedy mówimy o uczeniu sieci mamy zwykle na myśli konkretny algorytm uczenia



# Pamięć asocjacyjna

- Najprostszy algorytm uczenia pochodzi od Johna Hopfielda
- Sieć Hopfielda działa jak pamięć asocjacyjna – kojarzy obiekt z nim samym: dobieramy wagi tak, żeby na wyjściu otrzymać wejście
- Jest użyteczna wtedy, gdy na wejściu podamy niepełny albo zaszumiony bodziec – sieć Hopfielda odtwarza poprawny bodziec
- Przykład: OCR i skanowanie

# Sieć Hopfielda

- Sieć Hopfielda do rozpoznawania wzorców binarnych (+1/-1) o długości  $n$  to perceptron jednowarstwowy o  $n$  neuronach wejściowych i  $n$  neuronach wyjściowych. Funkcja aktywacji:  $f(x) = \text{sgn}(x)$
- Żeby zapamiętać zbiór wzorców  $P^1, P^2, \dots, P^m$ , gdzie  $k$ -ty wzorzec jest ciągiem liczb +/-1  
$$P^k = \{p_1^k, p_2^k, \dots, p_n^k\}$$
przyjmujemy wagi połączeń:

$$w_{i,j} = \frac{1}{n} \sum_{k=1}^m p_i^k p_j^k$$

# Sieć Hopfielda

- Dlaczego tak?

$$w_{i,j} = \frac{1}{n} \sum_{k=1}^m p_i^k p_j^k$$

- Dla każdego wzorca ( $P^k$ ) i dla każdego połączenia między neuronem wchodzącym  $i$  oraz wychodzącym  $j$  dodajemy do wagi człon  $1/n$ , jeżeli symbole na miejscu  $i$  oraz  $j$  są takie same, a odejmujemy  $1/n$ , jeżeli te symbole się różnią
- W ten sposób zapisujemy w macierzy wag informację o korelacjach w bodźcach.

# Sieć Hopfielda

- Metoda Hopfielda jest bardzo prosta, żeby jednak była skuteczna
  - Bodźce nie mogą być zbyt podobne do siebie
  - Bodźce wejściowe nie mogą być zbyt zniekształcone

- Jak nauczyć sieć czegoś nowego?

# Reguła uczenia perceptronu (PLR)

- PLR korzysta z par wejście – pożądane wyjście. Początkowo wszystkie wagi sieci wybieramy losowo, następnie iterujemy:
  - W każdej iteracji podajemy wzór wejściowy  $\{I_1, \dots, I_n\}$  ze zbioru uczącego.
  - Sieć generuje wyjście  $\{O_1, \dots, O_n\}$ , od którego odejmujemy pożądane wyjście  $\{D_1, \dots, D_n\}$  otrzymując wektor błędu  $\{E_1, \dots, E_n\}$ .
  - Długość tego wektora jest miarą sukcesu sieci. Jeżeli wynosi 0, to znaczy, że sieć dała żądany wynik dokładnie. Im większy błąd, tym gorzej poradziła sobie sieć.

# Reguła uczenia perceptronu (PLR)

- Każdą wagę sieci zmieniamy w zależności od wartości składowej błędu, z którą jest związana.

- Zmodyfikowana waga przyjmuje wartość

$$w_{i,j} + \eta I_i E_j$$

gdzie  $\eta$  jest parametrem szybkości uczenia. Powinien on wynosić ok. 0,1. Jeżeli jest za duży, wówczas sieć szybko „zapomina” to, czego się nauczyła. Jeżeli za mały, to sieć uczy się bardzo wolno.

# Reguła uczenia perceptronu (PLR)

- Wielokrotne powtarzanie tej reguły poprawia wyniki do pewnego punktu – błąd zwykle nie zbiega do zera
- Jest to związane z generalizacją sieci – musi działać dobrze także dla wzorców, których nie widziała.



# Przykład: rozpoznawanie liter

- Program **Hopfield**

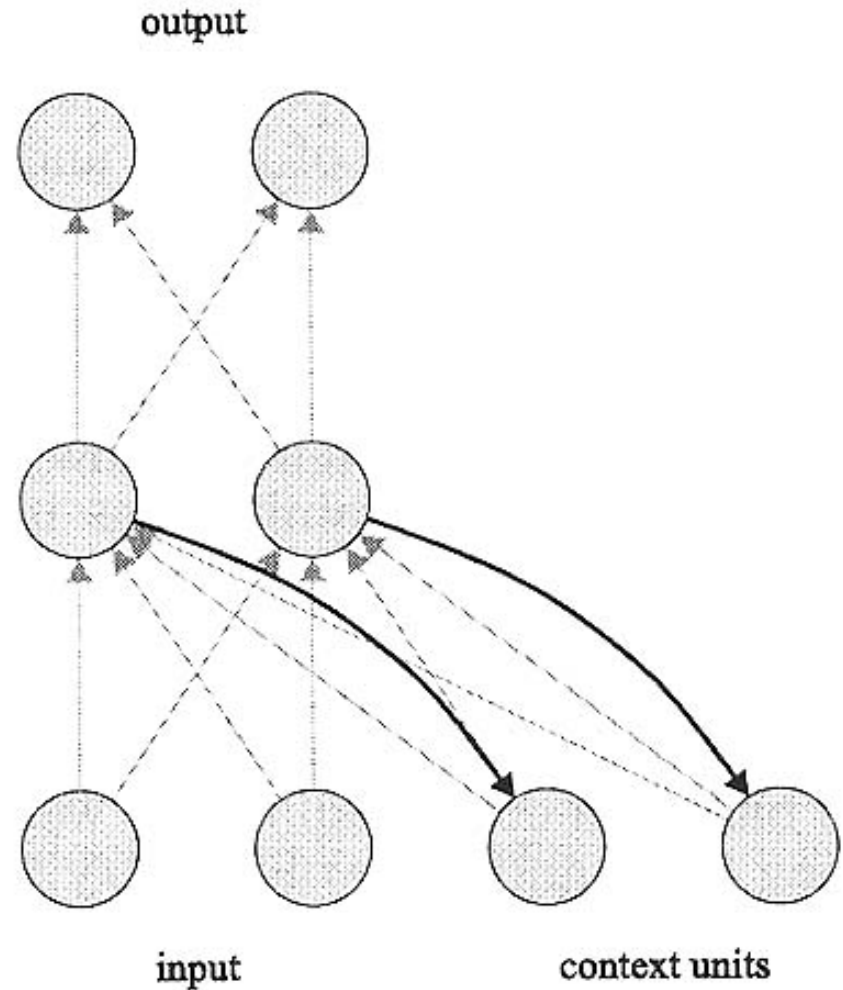
# Metoda propagacji wstecznej

- Najpopularniejsza metoda uczenia perceptronów wielowarstwowych
- Uogólnienie reguły uczenia perceptronu
- Zmieniamy wagi począwszy od warstwy wyjściowej cofając się do wejścia
- Zmiany wag zachodzą zgodnie z PLR

# Sieci rekurencyjne

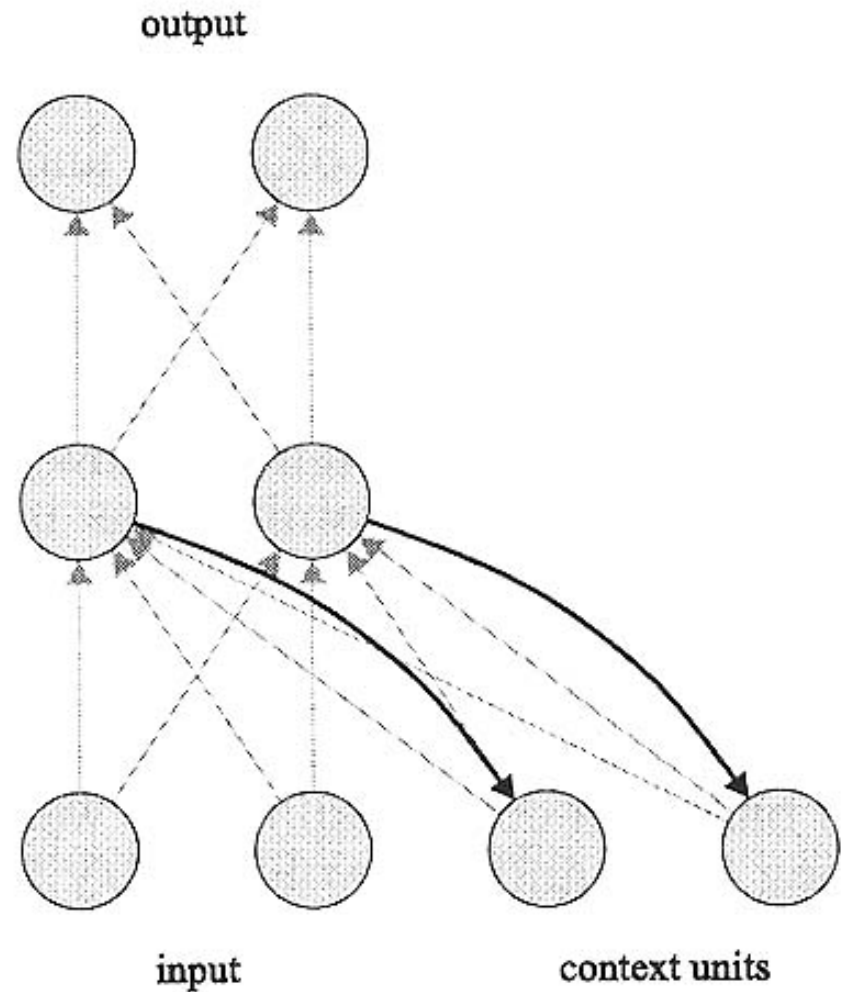
- Uogólnienie sieci jednokierunkowych.
- Sieć Elmana: pamięta poprzednie wejście

dwuwymiarowy  
perceptron  
z dodatkowymi  
jednostkami –  
neuronami  
kontekstowymi



# Sieci rekurencyjne

- W odróżnieniu od sieci jednokierunkowych sygnał wychodzący z sieci rekurencyjnej zależy od historii
- Przykład: modelowanie indeksów giełdowych
- Uczenie: propagacja wsteczna (wagi sprzężeń zwrotnych są ustalone)



# Rodzaje uczenia sieci

- Uczenie pod nadzorem  
(wszystkie omawiane do tej pory)
- Uczenie ze wzmacnianiem:  
oceniaamy sieć na podstawie tego,  
jak sobie radzi z zadaniem
- Uczenie bez nadzoru: sieć nie dostaje  
żadnych wskazówek jak ma działać; na ogół  
takie sieci uczą się dzielić grupę sygnałów  
wejściowych na klasy sygnałów  
o „podobnym” charakterze (klasyfikacja).

# Podsumowanie

- Pomysł sieci neuronowych jest starszy od pomysłu komputera von Neumanna
- Rozwój teorii i praktyki sieci neuronowych odbywał się falami. Kluczowe momenty:
  - 1943: McCulloch i Pitts – sztuczny neuron
  - 1950': Rosenblatt – perceptrony
  - 1960': Minsky i Papert – krytyka perceptronu
  - 1980': sieci Hopfielda
- Obecnie: szerokie zastosowanie w wielu dziedzinach, m.in. do klasyfikacji i predykcji