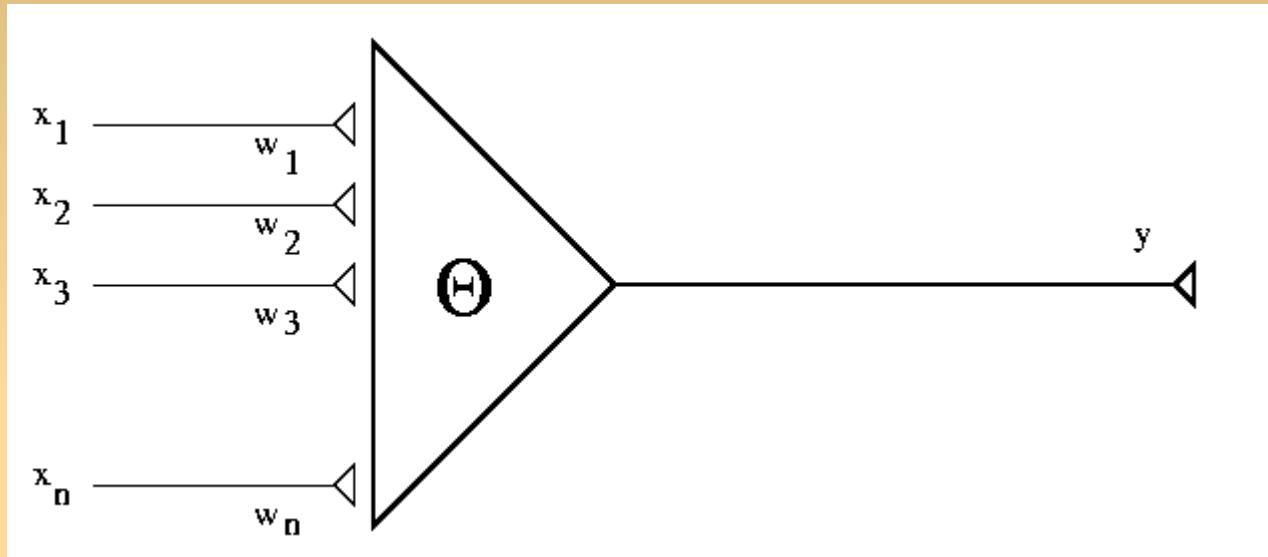


# Sieci neuronowe i ich ciekawe zastosowania



Autor: Wojciech Jamrozy  
III rok SMP / Informatyka

# Klasyczna algorytmika

- Sortowanie ciągu liczb
- Czy i ile razy dane słowo wystąpiło w tekście
- Najkrótsza droga pomiędzy dwoma miastami
- Czy istnieje para odcinków ze zbioru która się przecina?
- Ile jest liczb w danym przedziale/obiektów danego rozmiaru o jakiejś właściwości
- Numeryczne całkowanie

# Inne problemy

- Przewidywanie pogody
- Przewidywanie kursów giełdowych
- Rozpoznawanie przedmiotów na obrazie
- Rozpoznawanie mowy
- Tłumaczenie tekstu
- Rozpoznawanie chorób u pacjentów
- Określanie zdolności kredytowej
- Gra w Go/Szachy
- Sterowanie robotem w nieznanym środowisku
- Samochód bez kierowcy

# Konekcjonizm

- Konekcjonizm – modelowanie procesów myślowych za pomocą stanów pewnych obiektów (np. sztucznych sieci neuronowych i "wag" połączeń pomiędzy ich neuronami).
- Nie wiemy jak rozwiązać problem, ale wiemy jakiej klasy obiekt jest to w stanie zrobić

# Klasyfikacja i regresja

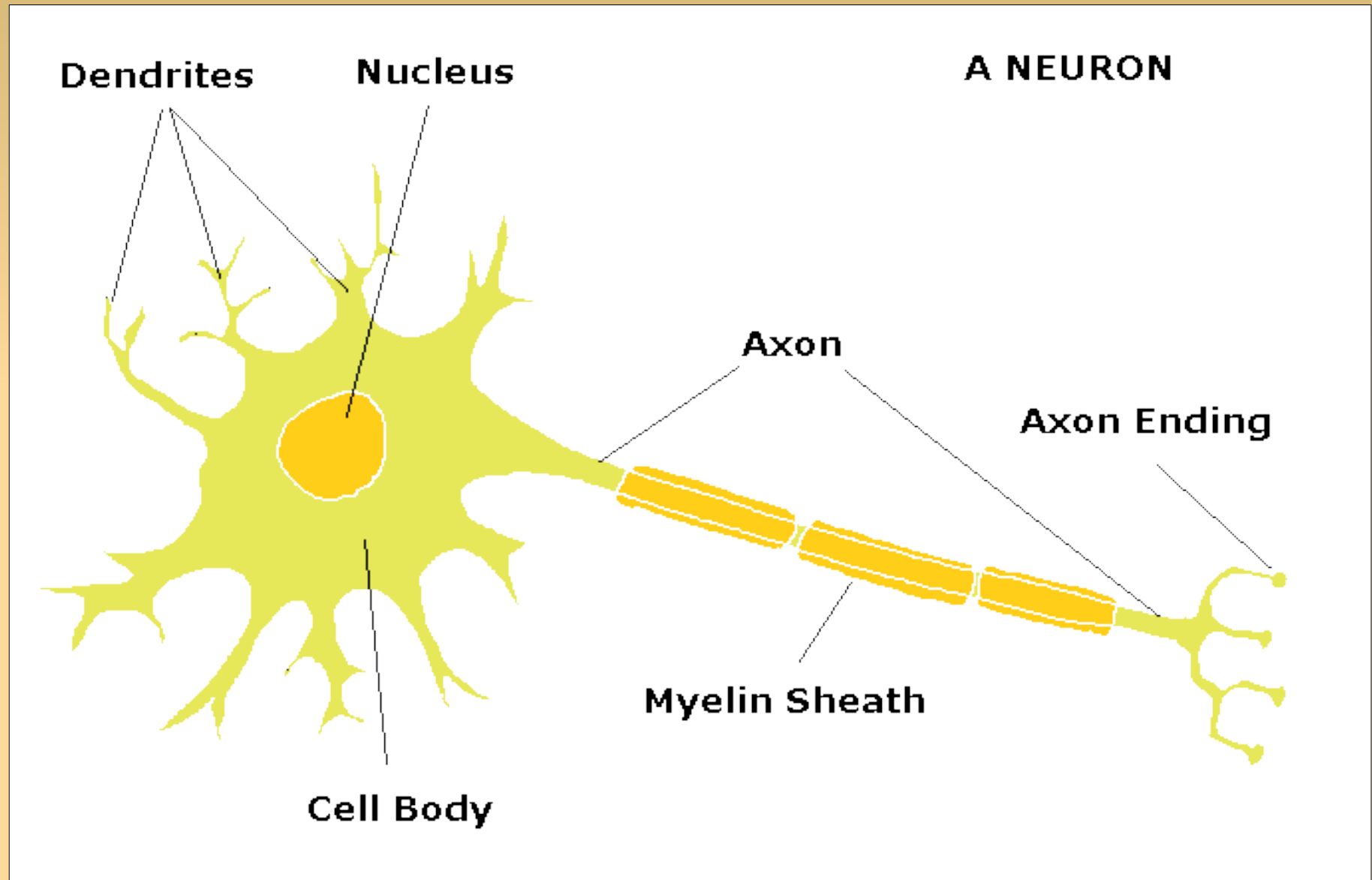
**Zbiór treningowy** – skończony zbiór par postaci  $(x,y)$  gdzie  $x$  to wektor atrybutów opisowych, a  $y$  to atrybut decyzyjny.  $\mathbf{X}$  i  $\mathbf{Y}$  są przestrzeniami z których pochodzą  $x$  i  $y$ .

Mając dany zbiór treningowy chcemy zbudować model, który wektorom z  $\mathbf{X}$  (również tym nieznanym) przypisuje najbardziej odpowiednie wartości z  $\mathbf{Y}$ .

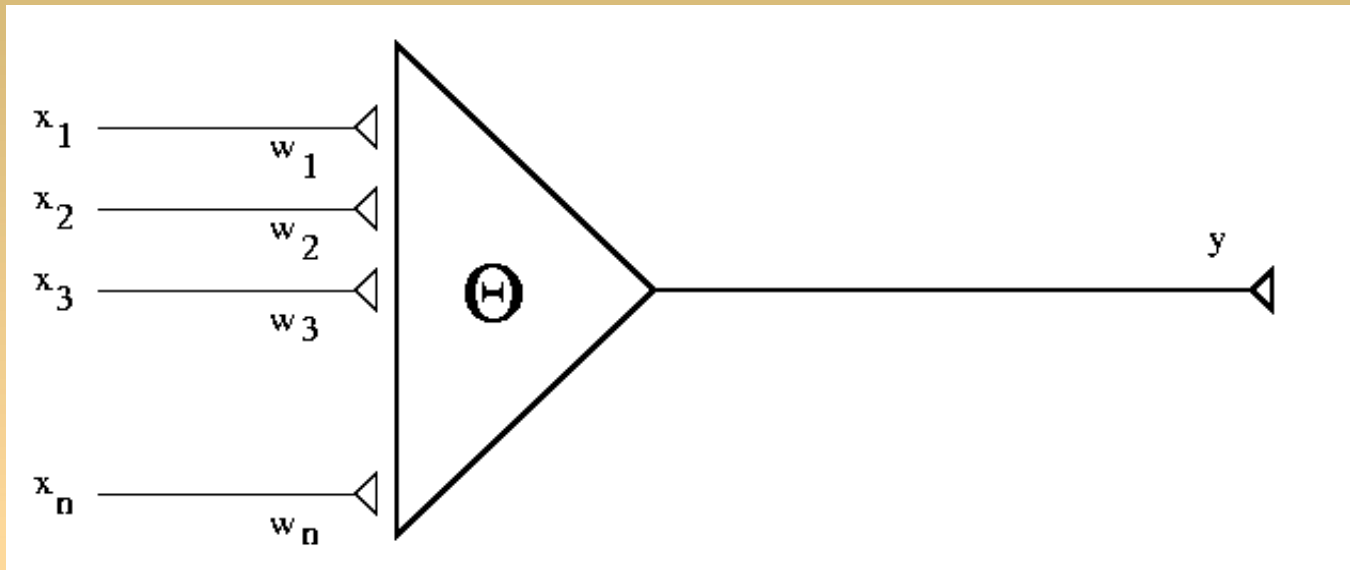
Gdy zbiór  $\mathbf{Y}$  skończony – **klasyfikacja**, w przeciwnym wypadku **regresja**

**Nauczanie z nauczycielem i bez nauczyciela**

# Neuron biologiczny



# Neuron sztuczny



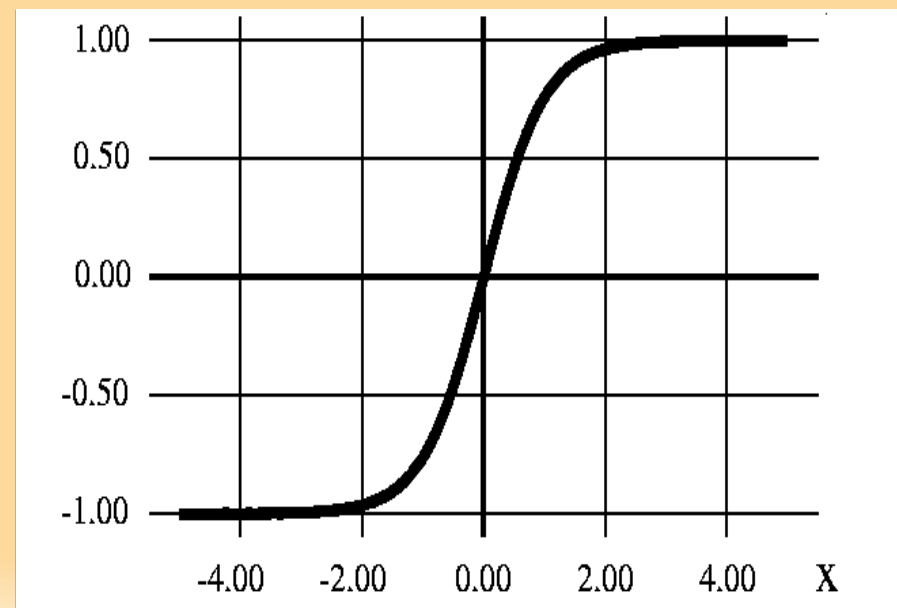
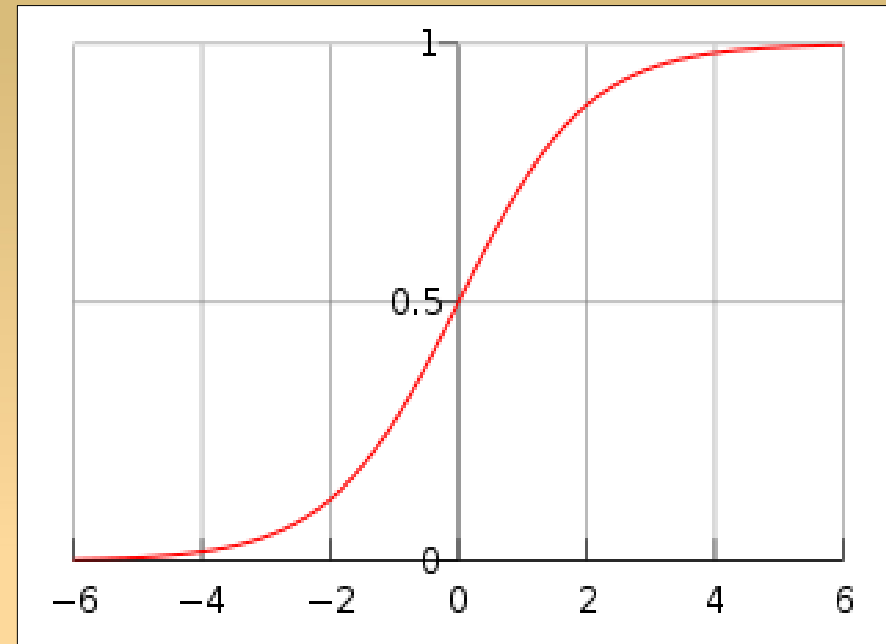
$$v = w_0 + \sum_{k=1}^n w_k x_k$$
$$y = \Theta(v)$$

# Funkcje aktywacji

$$\Theta(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases}$$

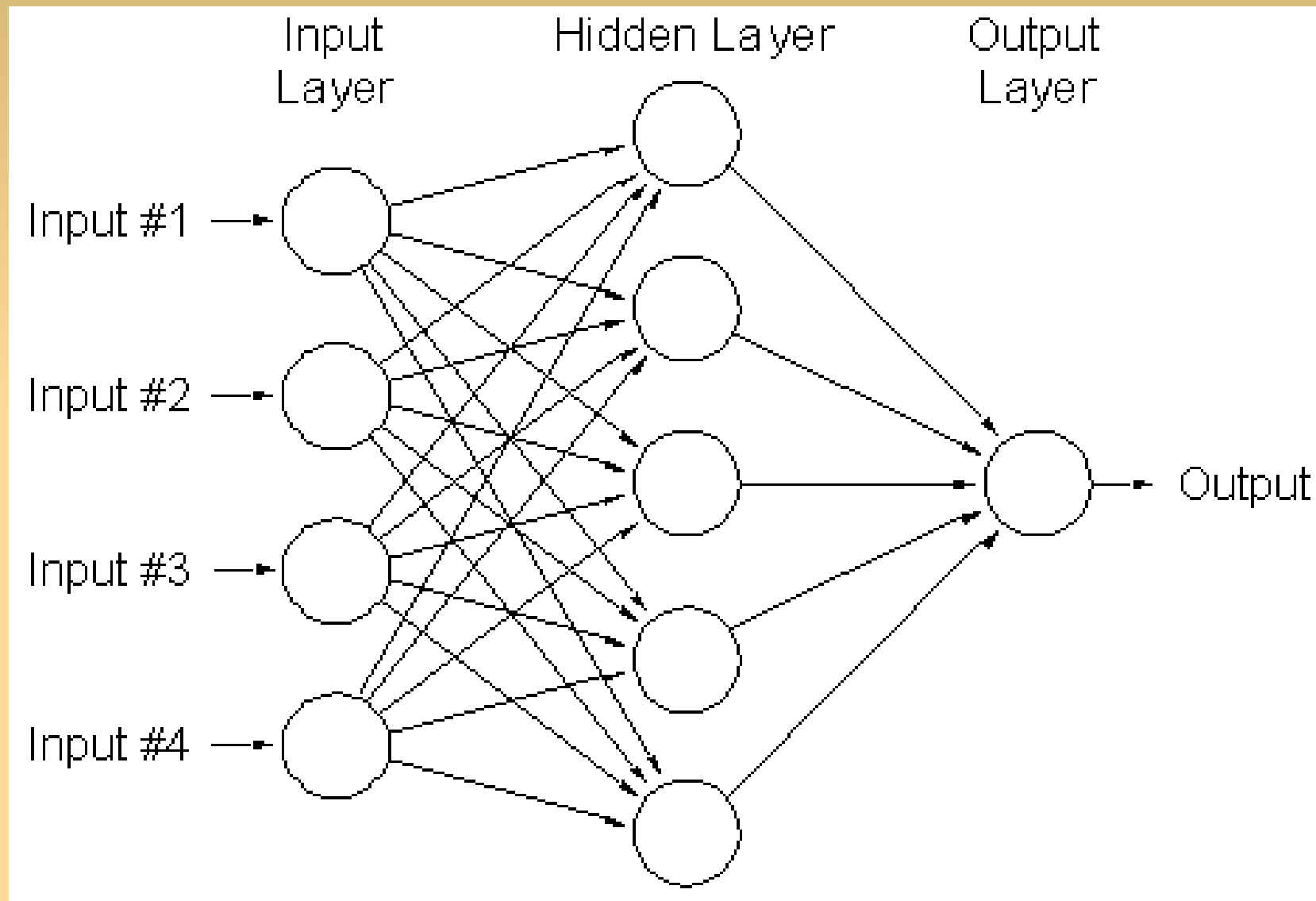
$$\Theta(v) = \frac{1}{1 + e^{-v/s}}$$

$$\Theta(v) = \tanh(v/s) = \frac{1 - e^{-v/s}}{1 + e^{-v/s}}$$





# Sieci warstwowe



# Backpropagation

$$e_k = d_k - y_k$$

$$E = \frac{1}{2} \sum_k e_k^2$$

$$v_k = \sum_{j=0}^n w_{kj} x_j$$

$$y_k = \varphi(v_k)$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial e_k} \frac{\partial e_k}{\partial y_k} \frac{\partial y_k}{\partial v_k} \frac{\partial v_k}{\partial w_{kj}}$$

$$\frac{\partial E}{\partial e_k} = e_k$$

$$\frac{\partial e_k}{\partial y_k} = -1$$

$$\frac{\partial y_k}{\partial v_k} = \varphi'(v_k)$$

$$\frac{\partial v_k}{\partial w_{kj}} = x_j$$

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta e_k \varphi'(v_k) x_j$$

- Jak nauczamy
- Generalizacja
- Przeuczenie

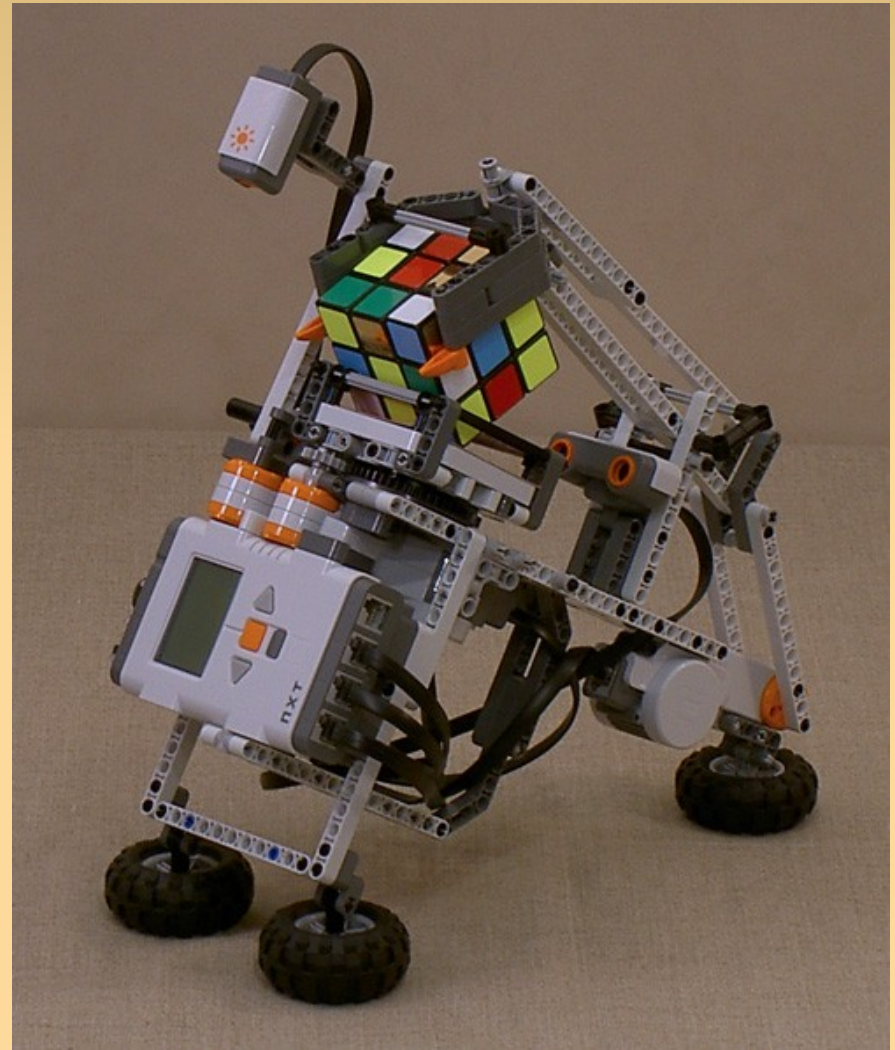
# Cechy potomstwa zwierząt

- Pokrewieństwo
- Laktacje: ilość, długość, masa, tłuszcz i białko
- Matka, ojciec, stado, rasa
- Chcemy przypisać potencjalną kozę do jednej z czterech kategorii (mleko, sery twarde i miękkie, inne)



# Kolory w robocie Lego

- Odczyt R, G, B z czujnika
- Silne zaszumienie i wrażliwość na światło
- Chcemy określić jaki mamy kolor (np. Czerwony, niebieski, biały)



# Jazda samochodem

Autonomous

Land

Vehicle

In a

Neural

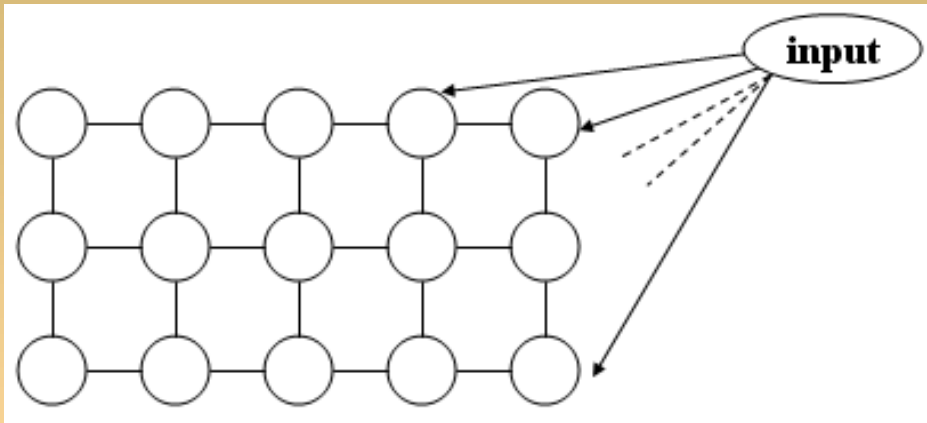
Network

(filmiki)

# Sterowanie humanoidem

- Natural Motion (spin-off Oxfordu)
- Wiele filmów i gier
- (filmik)

# Sieć Kohonena



- Każdy neuron zawiera prototyp z przestrzeni wejściowej
- "Wygrywa" neuron najbliższy położony aktualnemu przykładowy
- Neurony bliskie na siatce są bliskie w pełnej przestrzeni
- Sieć jest ekstrakcją cech przestrzeni wejściowej

# Wizualizacja danych

- Sieci Kohonena możemy użyć do wizualizacji danych wielowymiarowych
- Przykłady (programy w R):
- Losowe punkty z sześcianu  $1 \times 1 \times 1$
- Losowe punktu z płaszczyzny o nierównomiernej gęstości



# Kompresja

- Obraz dzielimy na fragmenty (np. 5x5)
- Traktujemy je jako wektory w przestrzeni 25-wymiarowej
- Uczymy na tych danych sieć Kohonena
- Zastępujemy fragment obrazu najbliższym położonym prototypem

# Kompresja Leny



# Kompresja Leny



# Kompresja Leny



# Sieci RBF

Funkcja RBF:  $\varphi(t) = e^{-\frac{t^2}{m}}$

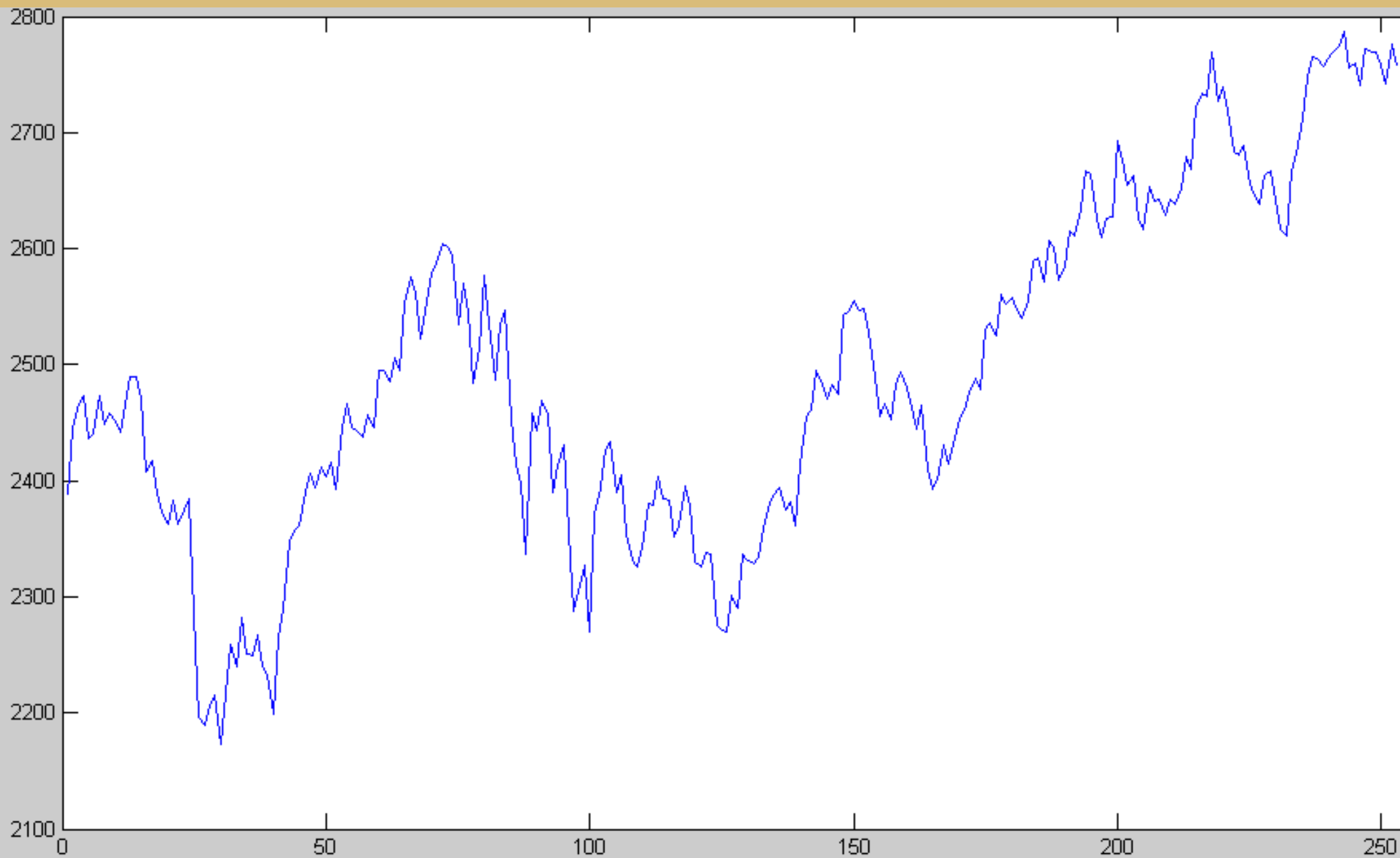
Mamy zbiór:  $(x_i, y_i) \quad i = 1, \dots, N$

Szukamy:  $F(x) = \sum_{i=1}^N w_i \varphi(\|x - x_i\|)$

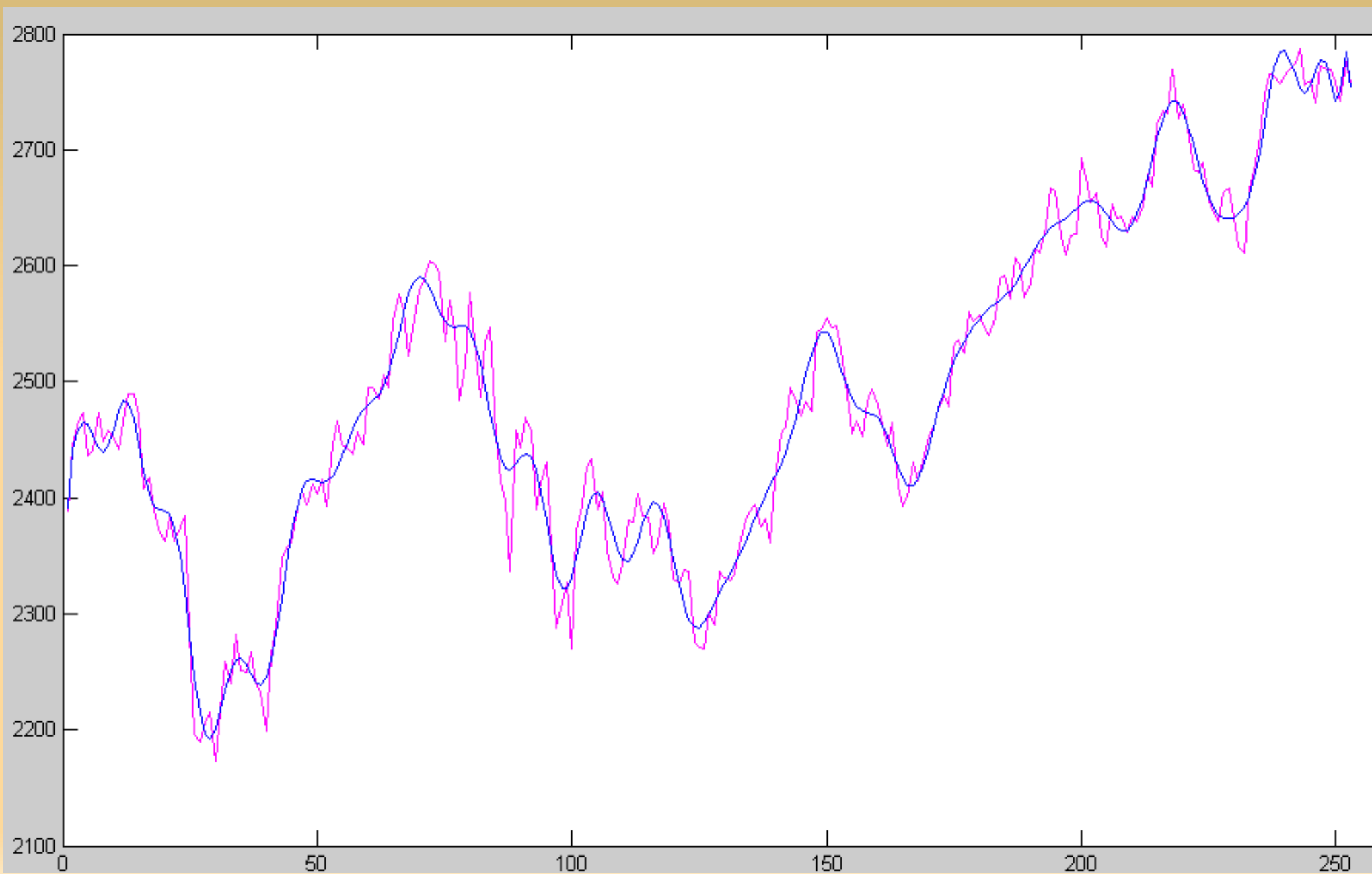


- Aproksymacja funkcji przez funkcję gładką

# Aproksymacja funkcji



# Aproksymacja funkcji



# Odrobina historii

- 1943 - McCulloch (psychiatra i neuroanatom) i Pitts (matematyk) podają pierwszy model sztucznego neuronu
- 1949 - Donald Hebb opisuje fizjologiczną regułę nauczania
- 1958 - Rosenblatt wprowadza perceptron
- 1969 - Minsky i Papert udowadniają, że jednowarstwowy perceptron jest ograniczony do problemów separowalnych liniowo - zachamowanie rozwoju SN na wiele lat
- 1974 - Werbos opisuje algorytm wstecznej propagacji błędów - praca niezauważona
- 1982 - opisanie sieci Kohonena
- 1986 - ponowne "odkrycie" algorytmu wstecznej propagacji błędów przez Rumelharta
- 1988 - opisanie sieci RBF



# Implementacje

- C++/Python – OpenCV
- R – nnet/kohonen
- Matlab – NeuralNetwork toolbox
- Java - weka

# Bibliografía

- S. Haykin, Neural networks, a comprehensive foundation. Prentice Hall, 1999.
- J. Friedman T. Hastie, R. Tibshirani, The Elements of Statistical Learning. Springer Verlag, 2001
- Raúl Rojas, Neural Networks - A Systematic Introduction, Springer-Verlag, Berlin, 1996
- Russell, Norvig, Artificial Intelligence: A Modern Approach

# Dziękuję za uwagę

Czy są jakieś pytania?