Sztuczne sieci neuronowe

Marek Grochowski

Wydział Fizyki, Astronomii i Informatyki Stosowanej UMK

http://www.fizyka.umk.pl/~grochu/nn
grochu@is.umk.pl

2 czerwca 2023

Plan

1. Perceptrony, sieci wielowarstwowe jednokierunkowe (MLP)
2. Metody uczenia i algorytm wstecznej propagacji błędu
3. Radialne funkcje bazowe (RBF) i metody aproksymacji
4. Samoorganizacja, sieci SOM, uczenie konkurencyjne
5. Sieci dynamiczne: model Hopfielda, maszyny Boltzmana
6. Głębokie sieci neuronowe (DNN)
7. Sieci splotowe (CNN)
8. Sieci rekurencyjne (RNN) i uczenie sekwencji
9. Autoenkodery, wykrywanie cech, DBN
10. Sieci typu GAN

Literatura I

- Osowski S., *Sieci neuronowe w ujęciu algorytmicznym*, Wydawnictwo Naukowo-Techniczne, Warszawa 1996
- Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza RM, Warszawa 1993
- Ryszard Tadeusiewicz, Tomasz Gąciarz, Barbara Borowik, Bartosz Lepe, *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Polska Akademia Umiejętności, 2008
- J. Żurada, M. Barski, W., Jędruch *Sztuczne sieci neuronowe*, Wydawnictwo Naukowe PWN 1996

Literatura II


Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*. MIT Press, 2016, `http://www.deeplearningbook.org`

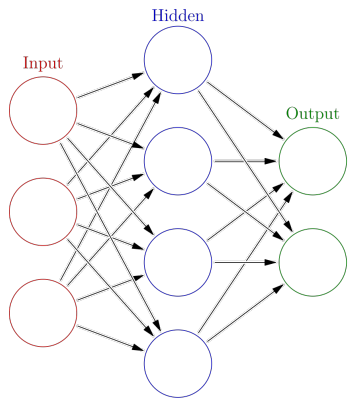

Michael Nielsen, *Neural Networks and Deep Learning*, `http://neuralnetworksanddeeplearning.com/`


Julien Vitay, *Neurocomputing*, `https://julien-vitay.net/lecturenotes-neurocomputing/`


Denny Britz, *Deep Learning Glossary*, `http://www.wildml.com/deep-learning-glossary/`

Sztuczne Sieci Neuronowe

Systemy obliczeniowe inspirowane biologicznymi sieciami neuronowymi składające się ze zbioru prostych jednostek przetwarzających sygnał (neuronów) komunikujących się między sobą za pomocą połączeń (wagi połączeń).



- model matematyczny $f(\mathbf{x}; \mathbf{W}) = \mathbf{y}$
- systemy uczące się, adaptowanie wag $\mathbf{W}$, Uczenie Maszynowe
- systemy inteligentne: Inteligencja Obliczeniowa, Sztuczna Inteligencja
- równoległe przetwarzanie sygnału

Inteligencja obliczeniowa

Computational Intelligence (CI) zajmuje się rozwiązywaniem problemów, które nie są efektywnie algorytmizowalne.

Cechą wielu systemów CI jest rozwiązywanie zadań na podstawie znanych przykładów, uczenie się z empirycznych danych zamiast programowania rozwiązania.

Problemy niealgorytmizowalne:

- zagadnienie jest zbyt złożone, np. problemy NP-trudne
- modelowany proces może zawierać trudne do zdefiniowania niejasności lub może być stochastyczny z natury
- procesu nie da się opisać przez zrozumiałe zasady
- warunki mogą się zmieniać, algorytm musi się dostosować do nowej sytuacji

Problemy efektywnie nieal gorytmizowalne

Problemy NP-trudne - Liczba kroków algorytmu dla złożonych sytuacji rośnie w sposób szybszy niż jakikolwiek wielomian liczby elementów (złożoności specyfikacji problemu).

Przykład: problem komiwojażera

Rysunek 1.1. Najkrótsza trasa premiera przebiegająca przez wszystkie miasta wojewódzkie w podziale administracyjnym z 1975 roku (A. Adrabiński)

Tabela 1.1. Czasy znalezienia przez komputer, wykonujący 100 miliardów operacji na sekundę, najkrótszej trasy podróży premiera (zob. ćwiczenie 1.3)

Liczba województw	Czas znajdowania najkrótszej trasy
17	3.5 minuty
25	$2 \cdot 10^5$ lat
49	$4 \cdot 10^{42}$ lat

Dla 100 miejsc mamy 100! (liczba 161 cyfrowa)

Rysunek: M. Sysło, „Algorytmy”

Problemy niealgorytmizowalne - przykłady

- rozumienie sensu zdań,
- działania twórcze, decyzje intuicyjne,
- rozpoznawanie twarzy i obrazów,
- rozpoznawanie pisma ręcznego,
- rozpoznawanie mowy i sygnałów, percepcja,
- sterowanie robotem, nieliniowymi układami,
- diagnostyka medyczna, planowanie terapii.

CI i sztuczna inteligencja (AI)

Klasyczne rozumienie AI i CI:
CI - percepcja i sterowanie: zachowania sensomotoryczne – sieci neuronowe i uczenie maszynowe
AI - wyższe czynności poznawcze: logika, język, rozumowanie, rozwiązywanie problemów.

Good old-fashon **Artificial Intelligence (GOFAI)** to część CI posługująca się symboliczną reprezentacją wiedzy, zajmuje się rozumowaniem, tworzeniem systemów ekspertowych.

CI i sztuczna inteligencja (AI)

Obecnie „sztuczna inteligencja” używana jest na określenie wszystkiego, co się kojarzy z inteligencją maszyn i algorytmów, które naśladują inteligentne zachowania człowieka.

- **Strong AI, Artificial General Intelligence** - agent świadomie działający i uczący sie działać w środowisku (podobnie do człowieka)
- **Weak AI** - maszyna rozwiązująca (ucząca sie rozwiązywać) konkretne zadanie

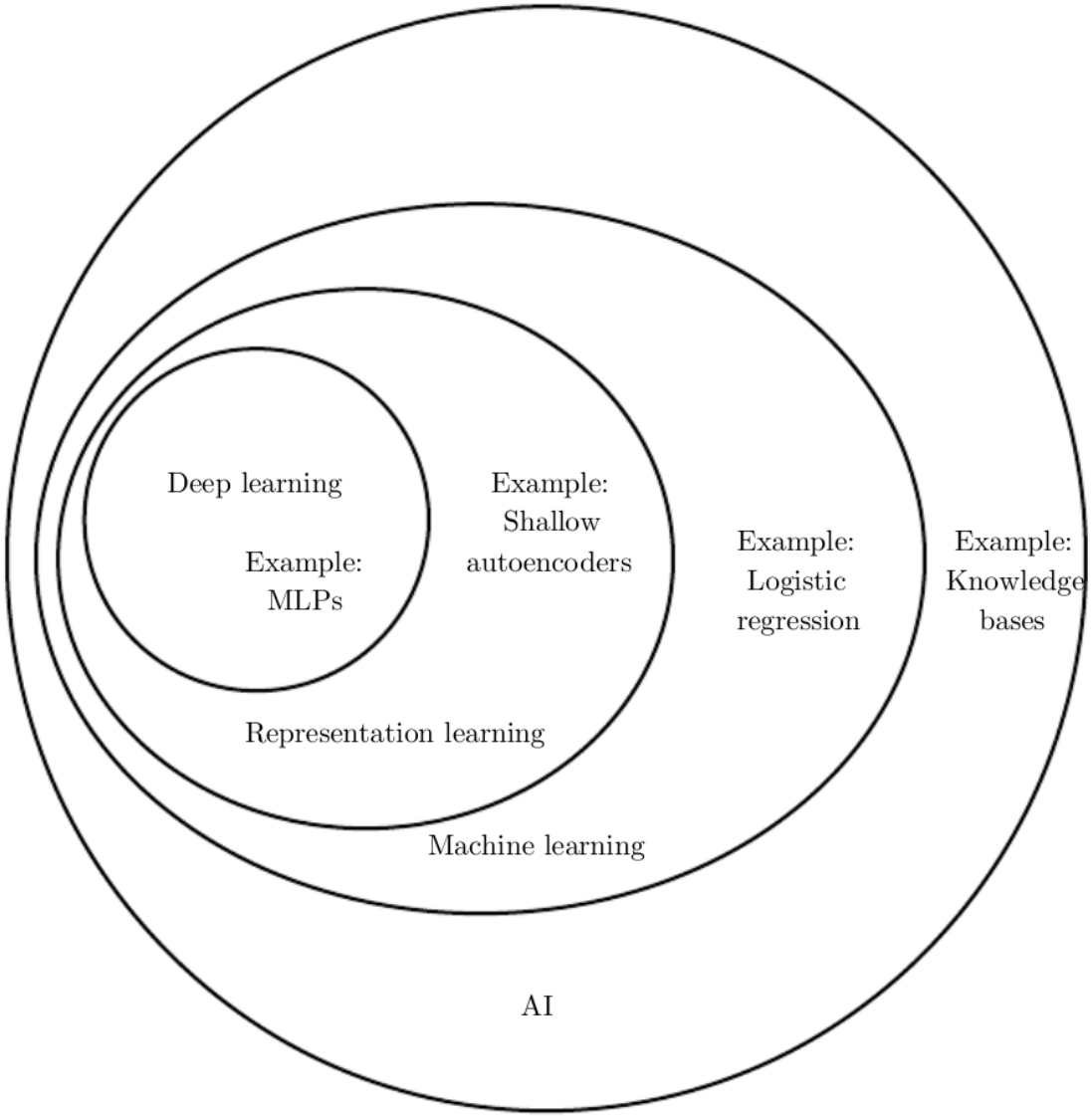
Efekt AI (twierdzenie Teslera)
„AI is whatever hasn’t been done yet”

Uczenie maszynowe

- **Uczenie maszynowe** (Machine learning, ML) - sytemy, które uczą się rozwiązywać problemy na podstawie dostarczonych przypadków uczących (zebranych danych)
- **Generalizacja** - dobrze wyuczony model działa poprawnie także dla przypadków, których nie było w danych treningowych
- Gdy dane się zmienią system można dostosować trenując go na nowych danych

Rysunek: https://xkcd.com/

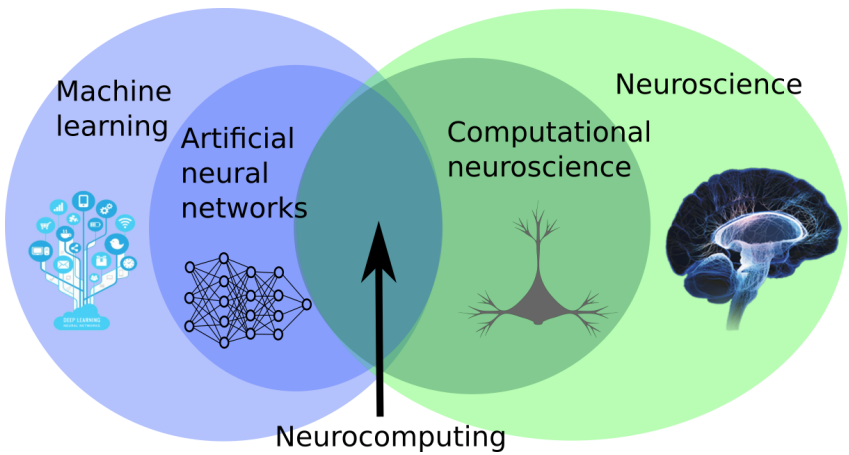
ANN i sztuczna inteligencja (AI)



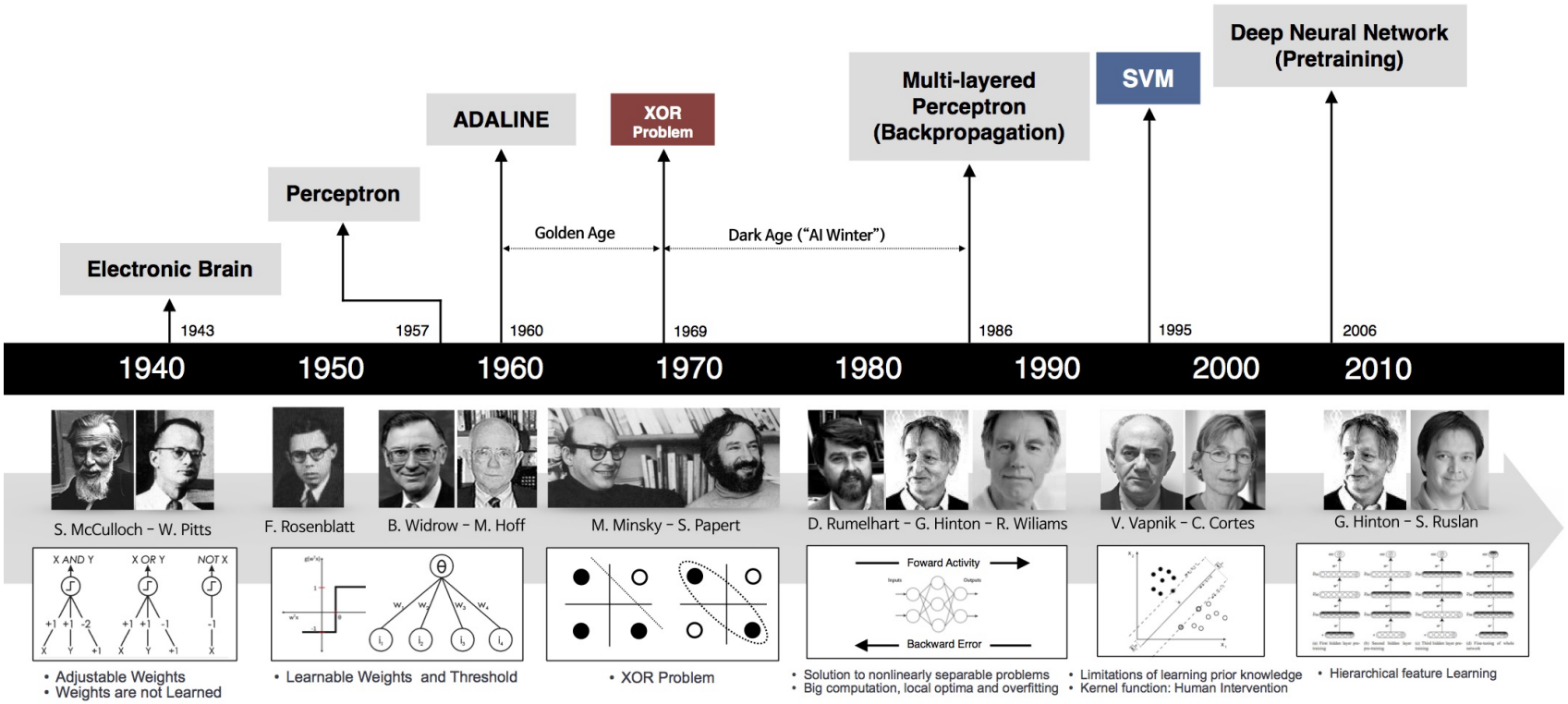
Gooffellow, 2016 [5]

Obliczenia neuronowe

- **modelowanie działania mózgu** - symulacje komputerowe, modele procesów kognitywnych
- przetwarzanie równoległe bardzo wydajne przy rozwiązywaniu niektórych problemów
- odporność na uszkodzenia, oddziaływania lokalne, uszkodzenie części sieci nie musi powodować drastycznych skutków
- rozwiązywanie praktycznych problemów za pomocą metod inspirowanych systemami biologicznymi
  -> **Sztuczne Sieci Neuronowe (Artificial Neural Networks)**



Zródło: J. Vitay [7]



https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html

Rys historyczny I

1938 N. Rashevsky, neurodynamika - sieci neuronowe jako układ dynamiczny

1943 W. McCulloch, W. Pitts, sieci neuronowe jako układy logiczne

1949 D. Hebb, uczenie się na poziomie synaptycznym (pierwsza reguła uczenia)

1952 A. Hodgkin, A. Huxley, szczegółowy matematyczny model neuronu kałamarnicy olbrzymiej (nagroda Nobla 1963)

1958 F. Rosenblatt, Perceptron - sieć jako funkcja matematyczna
J. von Neuman, The computer and the brain

1960 B. Widrow, M. Hoff, Adeline

1967 K. Steinbuch, E. Schmitt, Macierze uczące się

1969 M. Minsky, S. Papert, książka „Perceptrony" - wykazali ograniczenia sieci liniowych

1973 Chr. von der Malsburg, samoorganizacja w układzie nerwowym

Rys historyczny II

1982 J. Hopfield, model Hopfielda
T. Kohonen, samoorganizacja map topograficznych mózgu

1983 K. Fukushima, S. Miyake, T. Ito, Neokognitron, głęboka sieć neuronowa, inspiracja późniejszych sieci splotowych

1985 D. Ackley, G. Hinton, T. Sejnowski, maszyny Boltzmana

1986 D. Rumelhart, G. Hinton, R. Williams, wsteczna propagacja błędów

1987 G. Carpenter, S. Grossberg, model ART;
W. Freeman, Chaos w mózgu

1990 T. Poggio, F. Girosi, sieci RBF, teoria regularyzacji

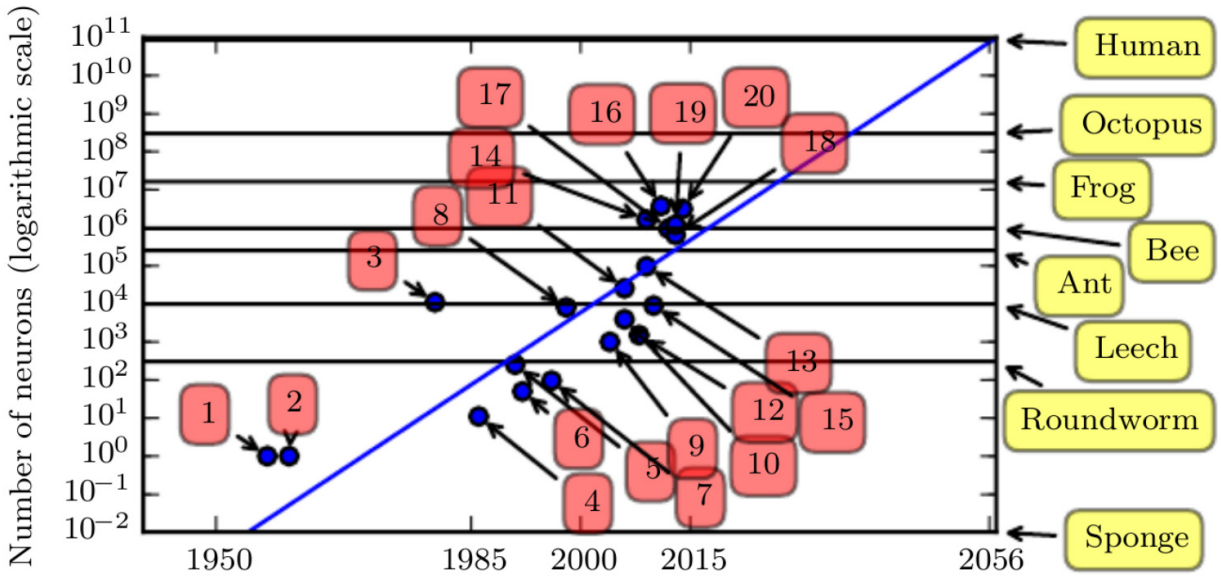
1995 V. Vapnik, SVM i „koniec ery Data Mining”

1997 S. Hochreiter, J. Schmidhuber, sieć rekurencyjna LSTM

1998 Y. LeCun, sieci splotowe w analizie obrazów

2005 GPU do sieci splotowych, bardzo duże sieci, głębokie uczenie

Ilość neuronów w modelach neuronowych



1. Perceptron (Rosenblatt, 1958, 1962)
2. Adaptive linear element (Widrow and Hoff, 1960)
3. Neocognitron (Fukushima, 1980)
4. Early back-propagation network (Rumelhart *et al.*, 1986b)
5. Recurrent neural network for speech recognition (Robinson and Fallsid
6. Multilayer perceptron for speech recognition (Bengio *et al.*, 1991)
7. Mean field sigmoid belief network (Saul *et al.*, 1996)
8. LeNet-5 (LeCun *et al.*, 1998b)
9. Echo state network (Jaeger and Haas, 2004)
10. Deep belief network (Hinton *et al.*, 2006)
11. GPU-accelerated convolutional network (Chellapilla *et al.*, 2006)
12. Deep Boltzmann machine (Salakhutdinov and Hinton, 2009a)
13. GPU-accelerated deep belief network (Raina *et al.*, 2009)
14. Unsupervised convolutional network (Jarrett *et al.*, 2009)
15. GPU-accelerated multilayer perceptron (Ciresan *et al.*, 2010)
16. OMP-1 network (Coates and Ng, 2011)
17. Distributed autoencoder (Le *et al.*, 2012)
18. Multi-GPU convolutional network (Krizhevsky *et al.*, 2012)
19. COTS HPC unsupervised convolutional network (Coates *et al.*, 2013)
20. GoogLeNet (Szegedy *et al.*, 2014a)

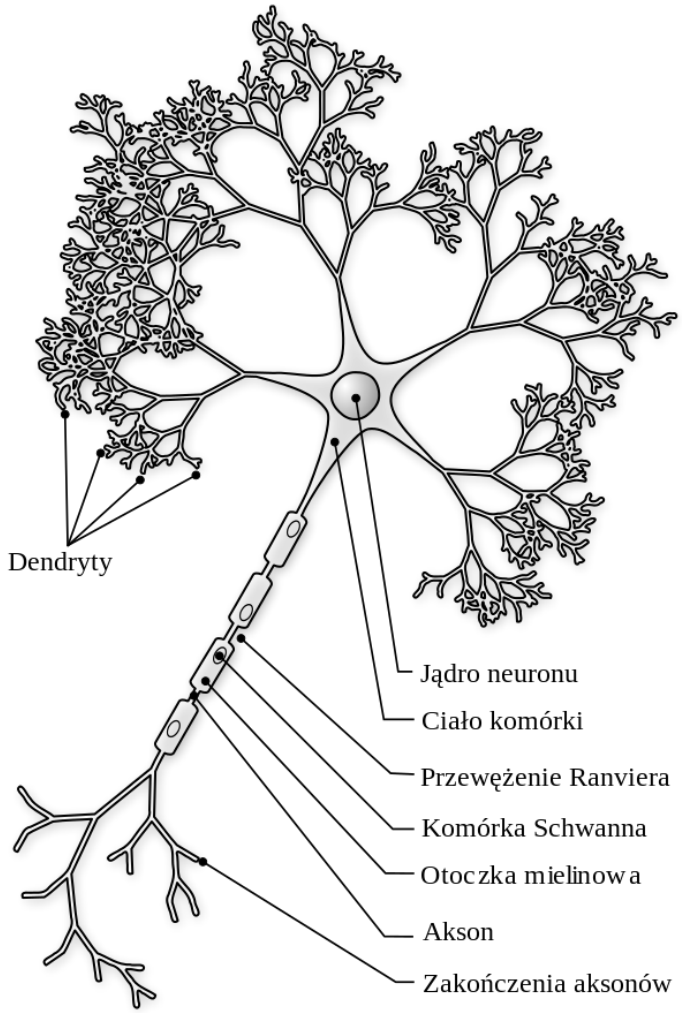
Liczba neuronów podwaja się co 2.5 roku

Gooffellow, 2016 [5]

Mózg i sieć neuronowa

- 10^{11} neuronów (100 miliardów), do 100Hz
- gęstość połączeń około 10^4 synaps na neuron
- Receptory - dostarczają sygnały wejściowe (bodźce wewnętrzne i sygnał ze zmysłów)
- Sygnał wyjściowy z układu nerwowego steruje efektorami (reakcja na bodziec)
- obliczenia równoległe
- układy biologiczne zbyt złożone do bezpośredniego modelowania - jedynie ogólne zasady organizacji i sposobu funkcjonowania układów biologicznych neuronów mogą służyć za inspirację
- Sztuczna siec neuronowa to model matematyczny, który można widzieć jako złożenie wielu (nieliniowych) funkcji i często nie ma nic wspólnego z sieciami biologicznymi

Neuron biologiczny



Rysunek: wikipedia.org

Neuron biologiczny

- **Dendryty** - wejście neuronu, odbierają sygnał od sąsiadujących neuronów, impulsy mogą być pobudzające lub hamujące
- **Ciało komórki (soma)**, akumuluje napływające w danym momencie sygnały
- **Akson** (wyjście), emituje sygnał, gdy potencjał w somie osiągnie odpowiedni próg
- **Synapsy** - styki pomiędzy dendrytami i aksonami, pobudzenie napływające z aksonu powoduje uwolnienie chemicznych neurotransmiterów, które wpływają na zachowanie połączeń neuronów
- Działanie neuronów biologicznych jest bardzo złożonym procesem, neurony w sztucznych sieciach są dużym uproszczeniem

☞ Schematic animation of spike generation in neural cell (J. Piersa)

Logika progowa neuronu

Neuron zbiera dochodzące do niego sygnały x_i obliczając sumę ważoną tych sygnałów (wagi w_i określane są przez procesy synaptyczne), tworząc z nich sumaryczny sygnał wejściowy:

$$I(t) = \sum_i w_i x_i(t)$$

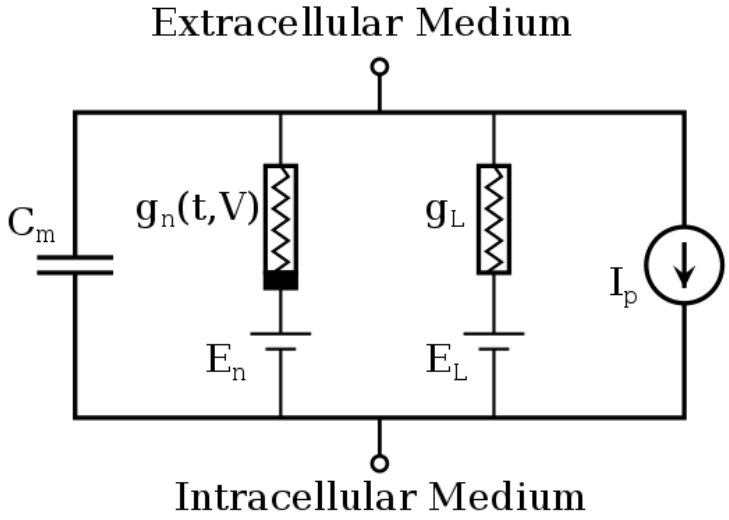
Sumowanie przestrzenne i czasowe potencjałów zgromadzonych na błonie komórkowej daje całkowitą aktywację elementu:

$$a(t) = F(I(t), a(t-1))$$

Aktywacja, razem z progiem wzbudzenia T , określa sygnał wyjściowy

$$o(t) = f(a(t), T)$$

Model Hodgkin–Huxley (1952)



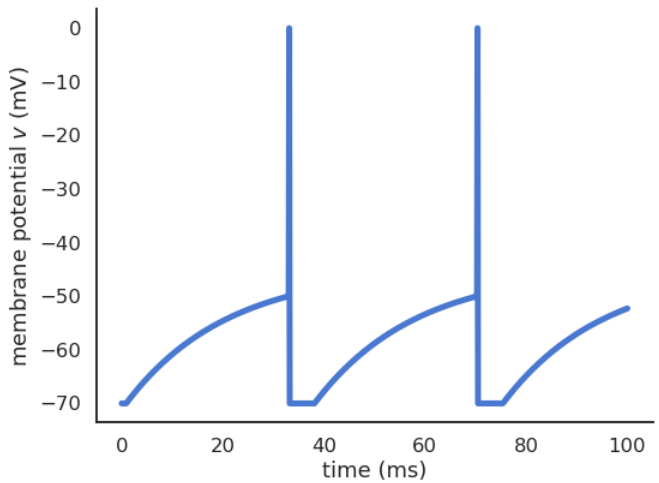
- model oparty na przewodnictwie
- zestaw nieliniowych równań różniczkowych, które przybliżają charakterystykę elektryczną komórek pobudliwych, takich jak neurony i komórki mięśniowe
- system dynamiczny w czasie ciągłym
- całkowity prąd przepływający przez membranę

$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + g_K \left(V_m - V_K\right) + g_{Na} \left(V_m - V_{Na}\right) + g_l \left(V_m - V_l\right)$$

https://en.wikipedia.org/wiki/Hodgkin-Huxley_model

Neurony impulsujace (*spiking neurons*)

Sumuj i odpal: *Leaky integrate-and-fire* (LIF; Lapicque, 1907) neurony generują impuls, gdy sumaryczny potencjał membrany przekroczy próg



$$C\frac{dv}{dt} = -g_L\left(v - V_L\right) + I$$

Gdy $v > V_T$ neuron emituje impuls a następnie odpoczywa

Źródło: Vitay, Neurocomputing [7]

Neurony kodowane częstością wzbudzeń

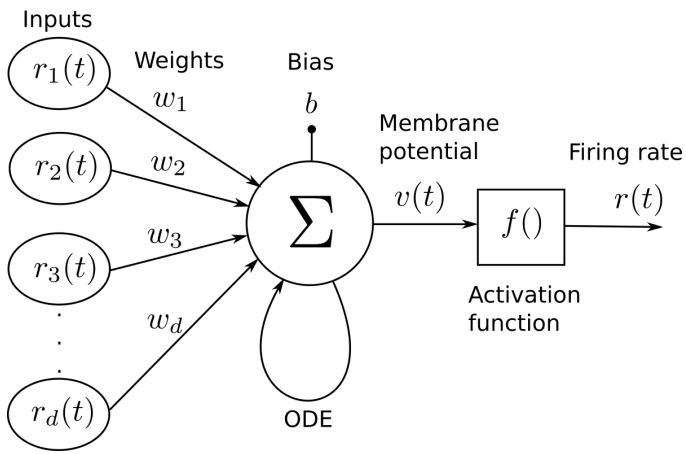
Rate-coded neurons

- potencjał membrany $v(t)$

$$\tau \frac{dv(t)}{dt} + v(t) = \sum_{i=1}^{d} w_{i,j} r_i(t) + b$$

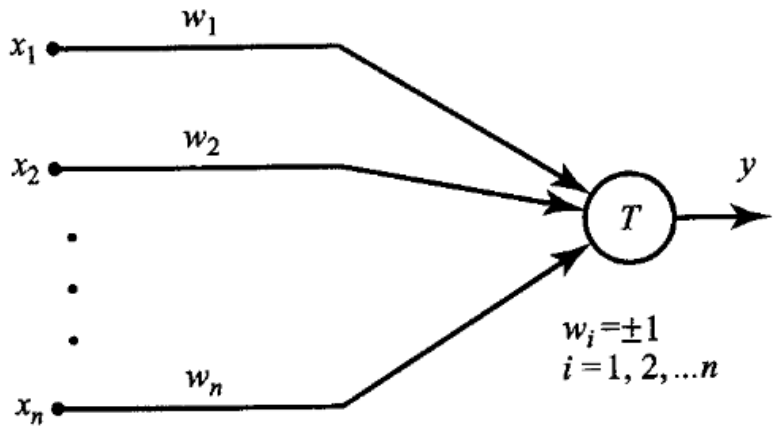
- częstość wzbudzeń (*firing rate*) $r(t)$ odwzorowuje potencjał membrany w pojedynczą wartość za pomocą **funkcji aktywacji**

$$r(t) = f(v(t))$$



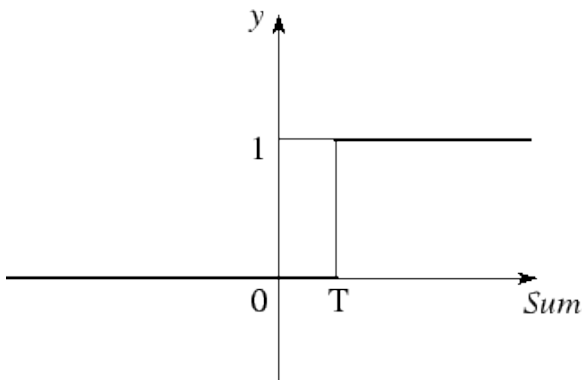
Źródło: Vitay, Neurocomputing [7]

Neuron McCulloch, Pitts (1943-49)



stan wyjściowy neuronu

$$o(\mathbf{x}) = \begin{cases} 1 & \text{gdy} \quad \sum_{i=1}^{n} w_i x_i \geq T \\ 0 & \text{gdy} \quad \sum_{i=1}^{n} w_i x_i < T \end{cases}$$



grafika: www.wikipedia.org

Neuron McCulloch, Pitts (1943-49)

- neurony mogą być tylko w dwóch stanach, aktywne albo nieaktywne, sygnał wejściowy i wyjściowy binarny

$$x_i \in \{0, 1\} \qquad o(\mathbf{x}) \in \{0, 1\}$$

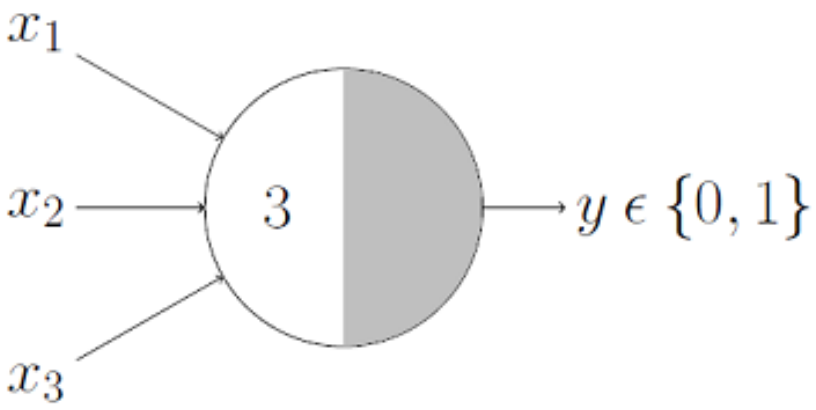
- sygnały dochodzą przez synapsy pobudzające i hamujące

$$w_{ij} \in \{-1, +1\}$$

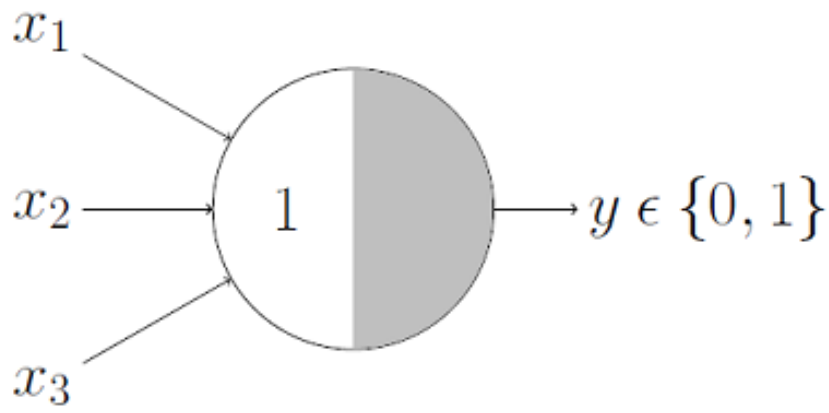
- każdy neuron ma ustalony próg pobudzenia $T > 0$
- sygnały sumują się w pewnym kwancie czasu i gdy przekroczą próg T to neuron się aktywuje
- brak uczenia się - wagi i próg mają stałą wartość
- odpowiedni dobór wag pozwala zrealizować funkcje logiczne: NOT, OR, AND bądź NOR i NAND -> sieć neuronów może zrealizować dowolną funkcję logiczną

Realizacja funkcji logicznych

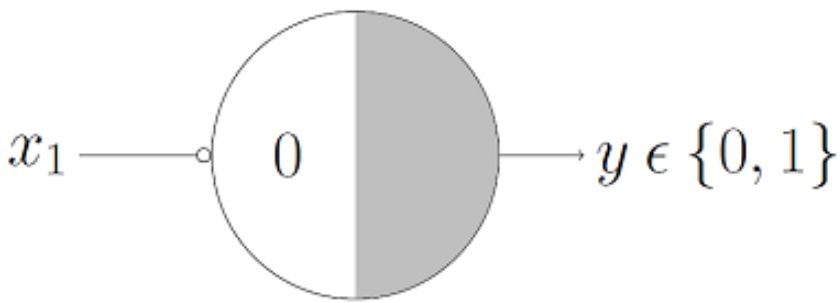
AND



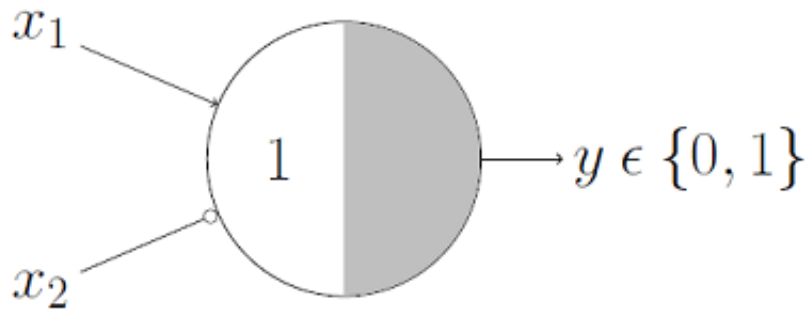
OR



NOT



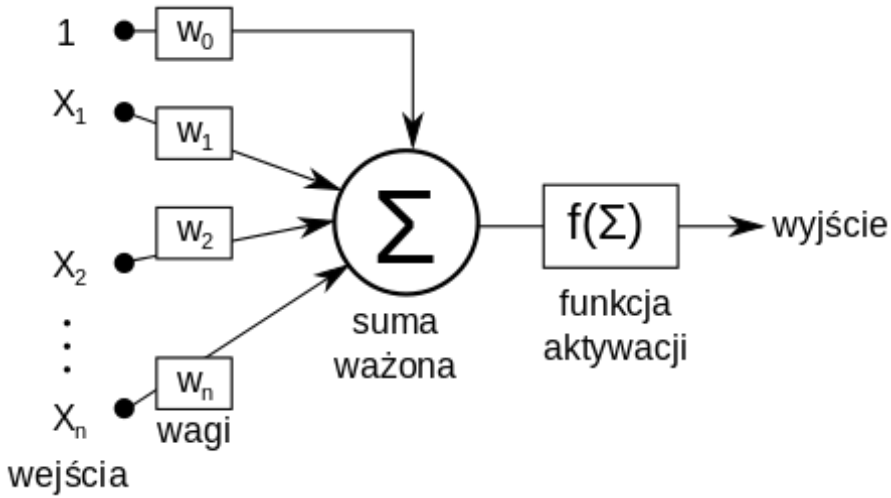
hamowanie sygnału wejściem x_2



$x_1\ AND\ !x_2$*

Źródło: Akshay L. Chandra, ☞ McCulloch-Pitts Neuron

Ogólny schemat neuronu



$$o(\mathbf{x}) = f\left(\sum_{i=1}^{n} w_i x_i + w_0\right) = f(\mathbf{w}^T \mathbf{x})$$

gdzie

$\mathbf{x} = [1, x_1, x_2, \ldots, x_n]$ wielowymiarowy sygnał wejściowy $\mathbf{x} \in \mathbb{R}^{n+1}$

$\mathbf{w} = [w_o, w_1, w_2, \ldots, w_n]$ wagi połączeń $\mathbf{w} \in \mathbb{R}^{n+1}$

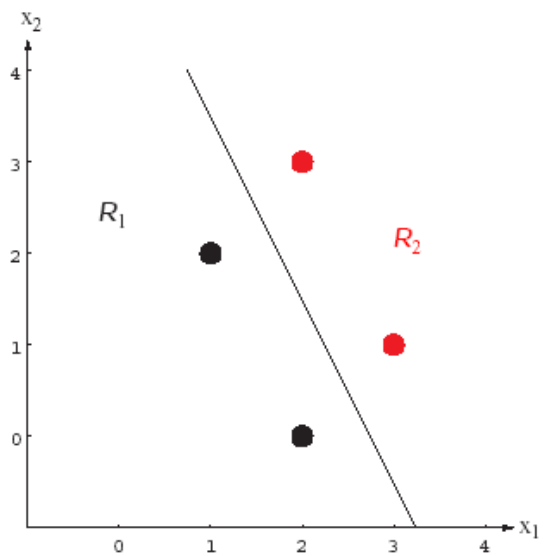
w_0 - wyraz wolny (bias), stały sygnał wejściowy $1 \cdot w_0$

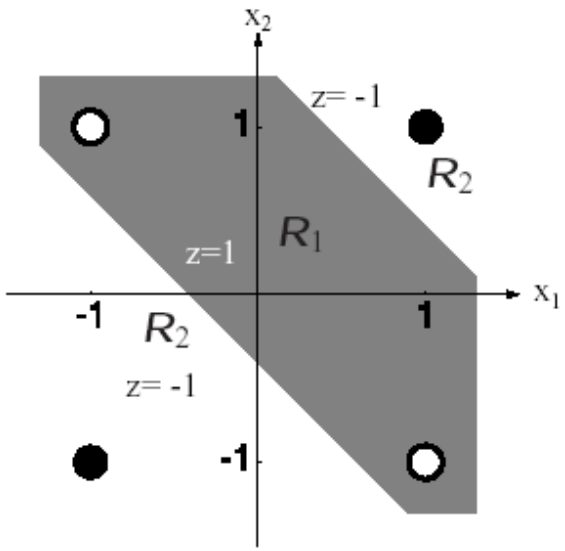
Interpretacja geometryczna - separowalność

Neuron z progiem realizuje separowalność liniową, hiperpłaszczyzna dzieli przestrzeń wejściową $\mathbb{R}^n$ na dwie części

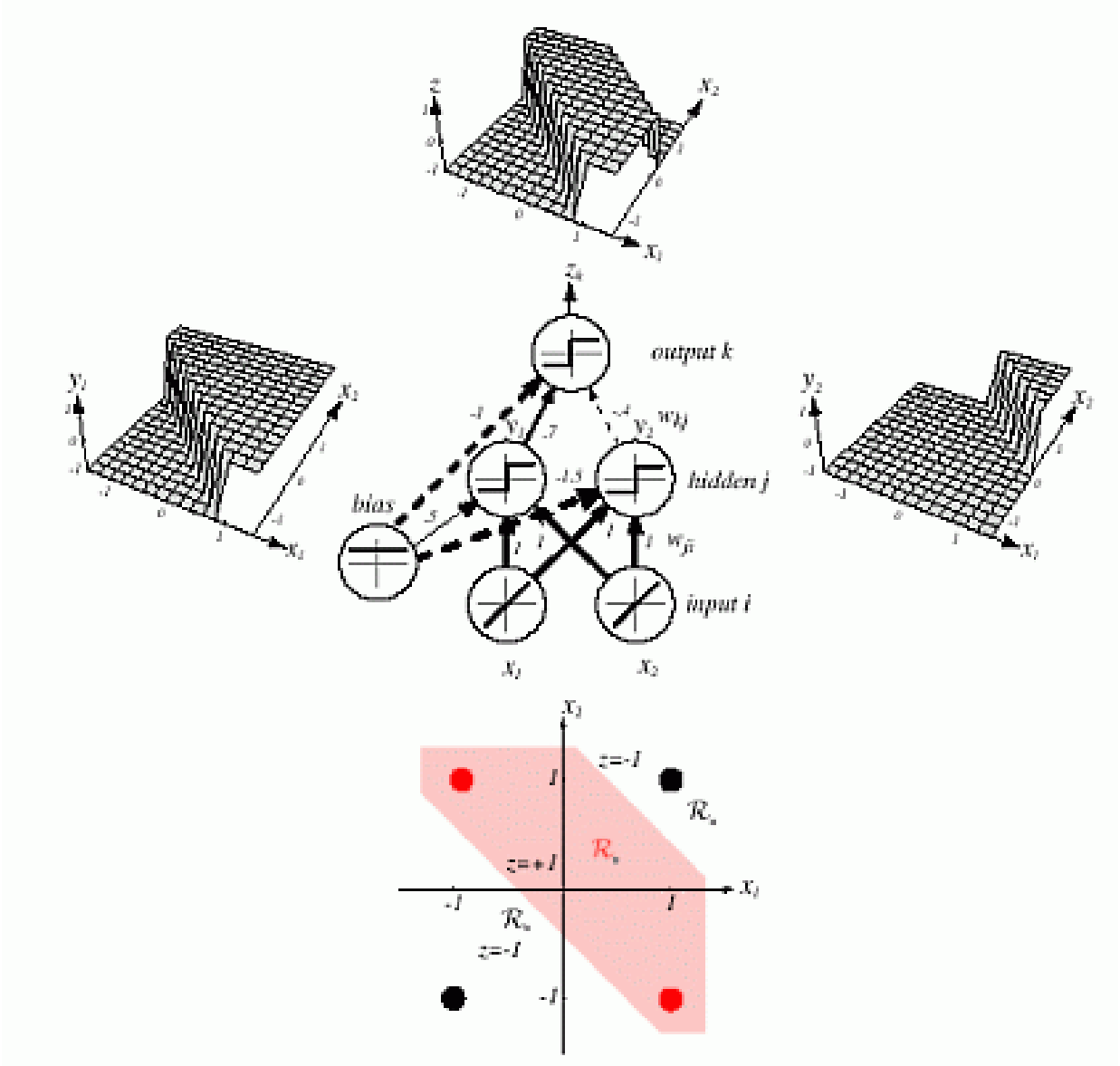
Przykład neuronu z 2 wejściami

$$o(\mathbf{x}) = \begin{cases} 1 & \text{gdy} \quad w_1 x_1 + w_2 x_2 \geq \theta \\ 0 & \text{gdy} \quad w_1 x_1 + w_2 x_2 < \theta \end{cases}$$





XOR



Reguła Hebba (1949)

$$\Delta w_i = \eta \cdot o(\mathbf{x}) \cdot x_i = \eta f(\mathbf{w}^\mathbf{T}\mathbf{x}) x_i$$

„Kiedy akson komórki A jest dostatecznie blisko by pobudzić komórkę B i wielokrotnie w sposób trwały bierze udział w jej pobudzaniu, procesy wzrostu lub zmian metabolicznych zachodzą w obu komórkach tak, że sprawność neuronu A jako jednej z komórek pobudzających B, wzrasta."

,,neurons that fire together wire together"

Plastyczność układu (uczenie)

Uczenie się: zmiana wag synaptycznych w_i w czasie

Reguła Hebba

$$\Delta w_i = \eta o(\mathbf{x}) x_i = \eta f(\mathbf{w}^\mathbf{T}\mathbf{x}) x_i$$

Reguła delta (Widrowa-Hoffa)

$$\Delta w_i = \eta (y - o(\mathbf{x})) x_i$$

gdzie y to spodziewana odpowiedź dla sygnału wejściowego $\mathbf{x}$

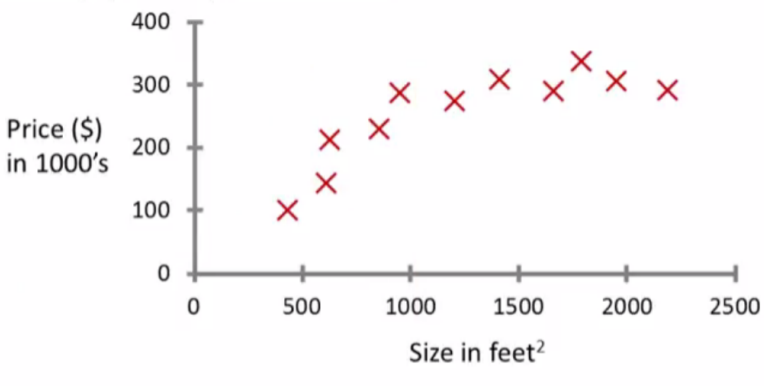
Rodzaje ucznia

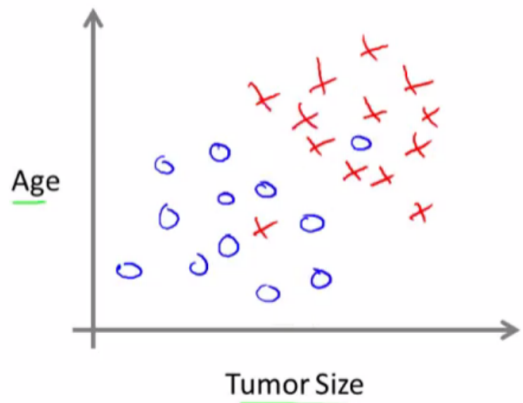
- **Uczenie bez nadzoru** (*unsupervised learning*)
  uczenie wyłącznie na podstawie sygnały wejściowego **X**, odkrywanie ciekawych struktur w przestrzeni danych wejściowych.
  Uczenie spontaniczne, odpowiada uczeniu w okresie niemowlęcym.
- **Uczenie nadzorowane** (*supervised learning*) - dla każdego sygnału wejściowego **x** dana jest pożądana odpowiedź **y**.
  Uczenie „szkolne".
- **Uczenie z krytykiem**, ze „wzmocnieniem" (*reinforcement learning*) - uczenie się zachowań, które przynoszą zysk po dłuższym czasie (np. autonomiczne roboty, gra z przeciwnikiem).
  Uczenie dojrzałe (nabieranie „mądrości").

Uczenie nadzorowane

Każdy przypadek **x** ze zbioru treningowego posiada przypisaną wartość **y**

- **Regresja (aproksymacja)** - obiekt wyjściowy **y** jest liczbą rzeczywistą lub wektorem liczb (np. temperatura)
- **Klasyfikacja** - obiekt wyjściowy **y** jest etykietą klasy (np. nazwa obiektu na zdjęciu)





Rysunek: Mohamad Ghassany, http://www.mghassany.com/MLcourse/introduction.html

Uczenie nadzorowane

- Model

$$f(\mathbf{x}; \mathbf{W}) = \mathbf{y}$$

  realizuje mapowanie zdefiniowane przez parametry sieci $\mathbf{W}$ (wagi połączeń) wektora treningowego $\mathbf{x}$ do wektora wyjściowego $\mathbf{y}$ (pożądanego wyjścia)
- Uczenie najczęściej polega na takim doborze parametrów $\mathbf{W}$ aby zminimalizować różnice pomiędzy wyjściem sieci $f(\mathbf{x})$ a wartością pożądaną $\mathbf{y}$, np.:

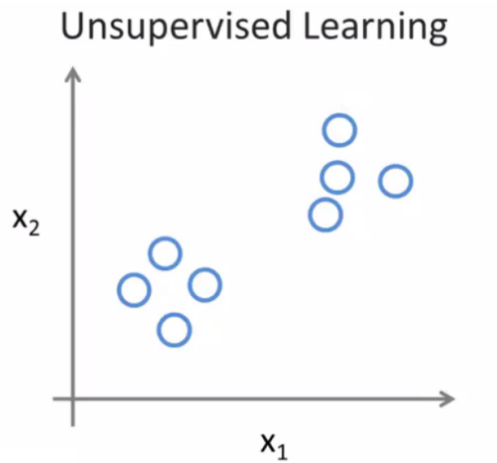
$$(f(\mathbf{x}; \mathbf{W}) - \mathbf{y})^2$$

- Celem uczenia nadzorowanego nie jest uczenie „na pamięć”, lecz **generalizacja**.

Uczenie nienadzorowane

Uczenie wyłącznie na podstawie sygnału wejściowego **X**, brak pożądanego sygnału wyjściowego. Przykłady:

- **klasteryzacja** (analiza skupień), automatyczne wykrywanie grup o podobnych właściwościach

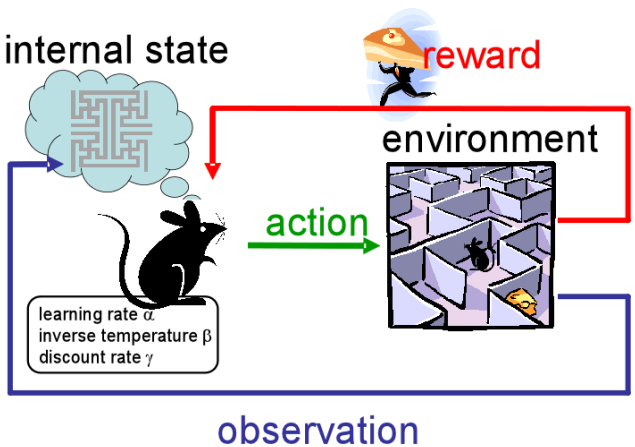


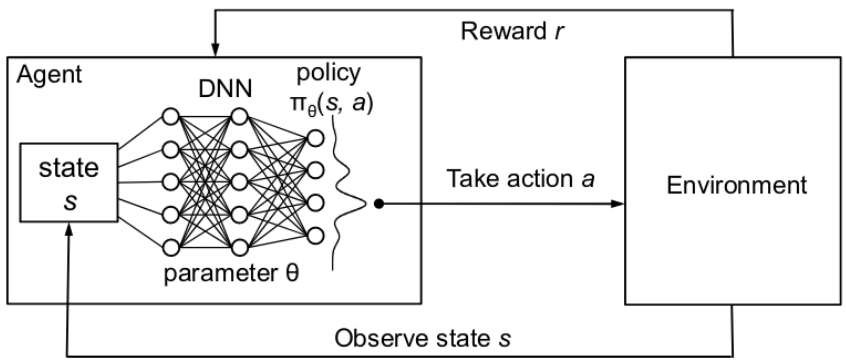
- wykrywanie cech i regularności, tworzenie przydatnych reprezentacji danych, kompresja (kodowanie) sygnału, redukcja wymiarowości, modelowanie sygnału, redukcja szumu

Rys.: Mohamad Ghassany, http://www.mghassany.com/MLcourse/introduction.html

Uczenie z krytykiem

- Sygnał wyjściowy jest zazwyczaj akcją (sekwencją akcji) podejmowanych przez agenta
- Nagroda (lub kara) jest dostarczana z opóźnieniem (np. przegrana/wygrana partia w szachy)

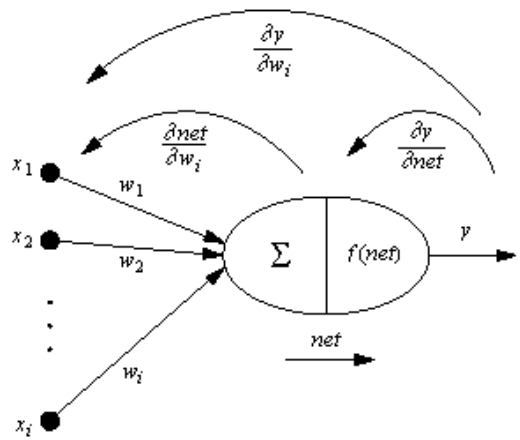




Rysunek: Hongzi Mao, „Resource Management with Deep Reinforcement Learning"
Aneek Das, The very basics of Reinforcement Learning

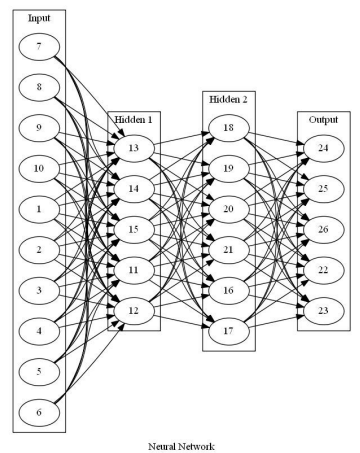
Główne aspekty modeli neuronowych I

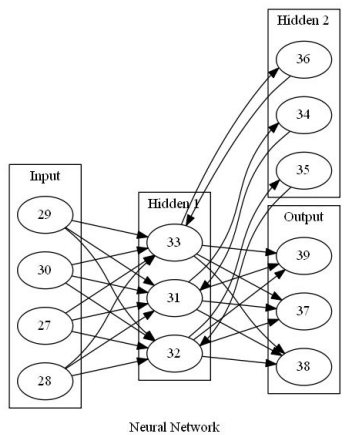
1. Sposób modelowania pojedynczego neuronu
   - stan aktywacji (wzbudzenia) poszczególnych neuronów, sposób aktywacji tych neuronów, funkcja opisująca sygnał wyjściowy elementu,
   - neurony: progowe, liniowe, nieliniowe (np. sigmoidalne)

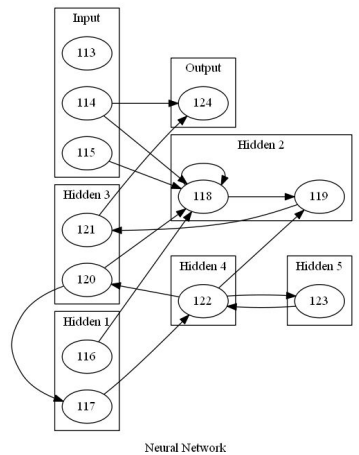


Główne aspekty modeli neuronowych II

2. Sposób propagacji sygnałów przez sieć neuronową
   - sieci jednokierunkowe (feedforward)
   - sieci ze sprzężeniami zwrotnymi, sieci rekurencyjne (recurrent), sieci dynamiczne
3. Topologia połączeń elementów (architektura sieci):
   - budowa warstwowa (warstwa wejściowa, wyjściowa, warstwy ukryte), struktura hierarchiczna
   - sieci jednowarstwowe, wielowarstwowe, płytkie, głębokie
   - warstwy w pełni połączone, rzadkie połączenia, splot







Główne aspekty modeli neuronowych III

4. Reguły uczenia - reguły modyfikacji parametrów opisujących neurony i połączenia pomiędzy nimi, algorytm optymalizacji, funkcja kosztu, regularyzacja
5. Otoczenie, w którym działa sieć neuronowa: sposób prezentacji danych, rodzaj danych wejściowych, reprezentacja sygnału wyjściowego (np. kodowanie etykiet klas)
6. Realizacja techniczna: język programowania, wykorzystanie GPU, zrównoleglenie obliczeń