

Sztuczne sieci neuronowe

Marek Grochowski

Wydział Fizyki, Astronomii i Informatyki Stosowanej UMK

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

2 marca 2025

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. Modele generatywne GAN, VAE
11. Transformery, mechanizm uwagi, LLM, ViT

Literatura I

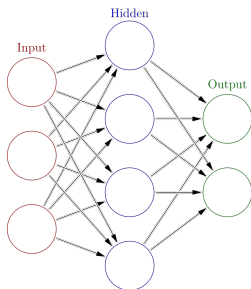
-  Osowski S., *Sieci neuronowe w ujęciu algorytmicznym*, Wydawnictwo Naukowo-Techniczne, Warszawa 1996
-  Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza RM, Warszawa 1993
-  Ryszard Tadeusiewicz, Tomasz Gąciarz, Barbara Borowik, Bartosz Lepe, *Odkrywanie właściwości sieci neuronowych 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/course-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 realizujący odwzorowanie $f(\mathbf{x}; \mathbf{W}) = \mathbf{y}$
- systemy uczące się, adaptowanie wag $\mathbf{W}$, Uczenie Maszynowe
- systemy inteligentne: Inteligencja Obliczeniowa, Sztuczna Inteligencja

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 niealgorytmizowalne

Problemy NP-trudne - złożoność obliczeniowa rośnie szybciej niż jakikolwiek wielomian liczby elementów

Przykład: problem komiwojażera



Rysunek 1.1. Najkrótsza trasa premiera przebiegająca przez wszystkie miasta województwie 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-fashion **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 (AGI)** - agent świadomie działający i uczący się działać w środowisku (podobnie do człowieka)
- **Weak AI** - maszyna rozwiązująca (ucząca się rozwiązywać) konkretne zadanie

Efekt AI (twierdzenie Teslera)

„AI is whatever hasn't been done yet”

Definicja AI jest elastyczna - granica między tym, co uważamy za AI, a tym, co jest już standardową technologią, ciągle się przesuwa.

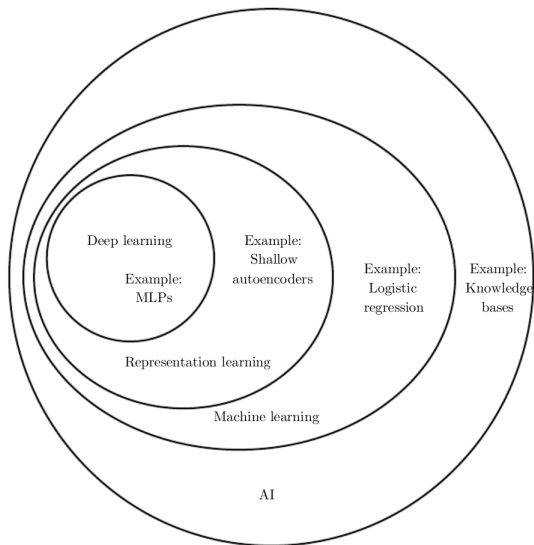
Uczenie maszynowe

- **Uczenie maszynowe** (Machine learning, ML) - systemy, 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 zmieniają system można dostosować trenując go na nowych danych



Rysunek: <https://xkcd.com/>

ANN i sztuczna inteligencja (AI)

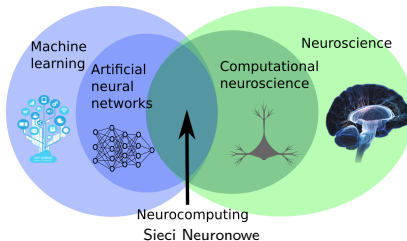


Gooffellow, 2016 [5]

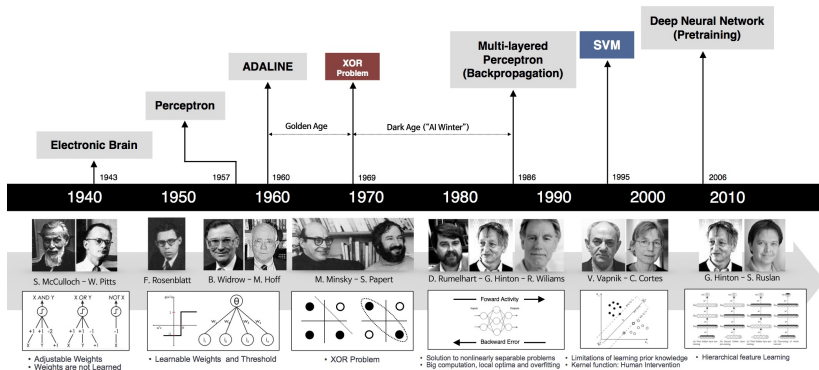
Obliczenia neuronowe

- **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

Wczesny etap rozwoju (1943-1960)

- 1938 N. Rashevsky, neurodynamika, sieci neuronowe jako układ dynamiczny (pierwszy model sieci neuronowej)
- 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

Pierwsze algorytmy uczące się (1960-1986)

- 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
- 1982 J. Hopfield, model Hopfielda
T. Kohonen, samoorganizacja map topograficznych mózgu
- 1985 D. Ackley, G. Hinton, T. Sejnowski, maszyny Boltzman
- 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”

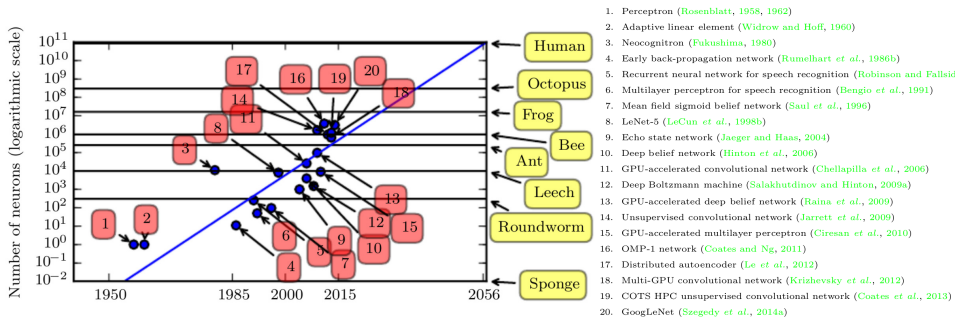
Początki głębokiego uczenia (1980-2010)

- 1983 K. Fukushima, S. Miyake, T. Ito, Neokognitron, głęboka sieć neuronowa, inspiracja późniejszych sieci spłotowych
- 1989 Yann LeCun, pierwsza sieć spłotowa LeNet
- 1997 S. Hochreiter, J. Schmidhuber, sieć rekurencyjna LSTM
- 1998 Y. LeCun, LeNet5, praktyczne zastosowanie CNN w analizie obrazów
- 2005 GPU do sieci spłotowych, bardzo duże sieci, głębokie uczenie
- 2006 Geoffrey Hinton, Deep Belief Networks (DBN)

Współczesny rozwój głębokiego uczenia (2010-)

- 2012 Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, sieć AlexNet pokonuje tradycyjne algorytmy w ImageNet (błąd 15%, wcześniej 30%), zastosowanie GPU, bez pre-treningu
- 2013 Kingma, Welling, variational autoencoder (VAE)
- 2014 Ian Goodfellow, generative adversarial networks (GAN)
- 2015 DeepMind, Deep Q-Network (DQN), deep RL, gry z Atari
- 2016 AlphaGo (DeepMind) pokonuje 19-to krotnego mistrza świata Lee Sedol w go
- 2017 Ashish Vaswani (Google Brain), transformery i rewolucja w przetwarzaniu języka naturalnego
- 2018 AlphaFold (DeepMind), przewidywanie struktury białek
- 2020 GPT-3 (OpenAI)
- 2021 OpenAI, generator obrazów DALL-E,
- 2022 Midjourney, Stable Diffusion, BARD/Gemini (Google AI)
- 2023 Llama (Meta), Grok (xAI)
- 2024 SORA (OpenAI) generowanie wideo, ...

Ilość neuronów w modelach neuronowych

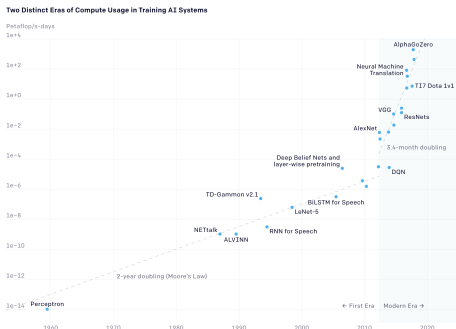


Prawo Moore'a dla sieci neuronowych:
liczba neuronów podwaja się co 2.5 roku

Goofellow, 2016 [5]

Era głębokiego uczenia

- Modern DL era: od 2012 roku, Alexnet, wraz z dostępnością wydajnego sprzętu (GPU, TPU) umożliwiające szkolenie coraz bardziej złożonych modeli z milionami, a nawet miliardami parametrów
- zmiana prawa Moore'a: podwojenie wydajności co 3.4 miesiąca

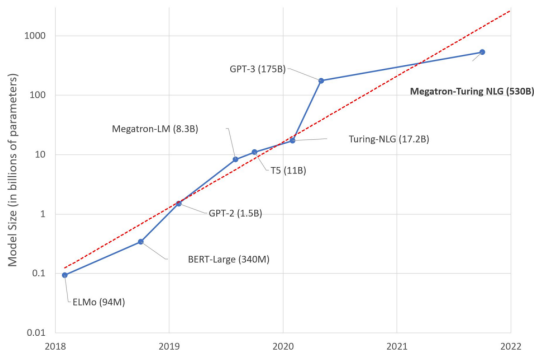


📊 Computation used to train notable artificial intelligence systems

📊 AI and Scaling the Compute for the new Moore's Law

Era dużej skali

- Large scale era: od 2016 roku, pojawienie się AlphaGo, masowe zrównoleglenie obliczeń, przetwarzanie wielkich paczek, algorytmy wyszukiwania architektur (NAS), expert iteration (ExIt)



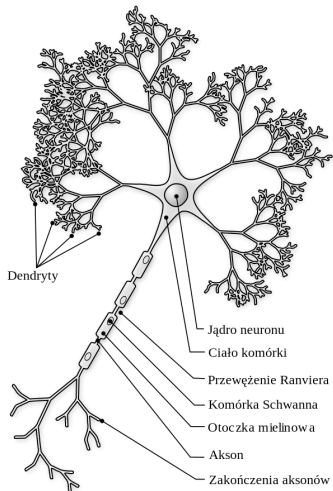
📊 Computation used to train notable artificial intelligence systems

📊 AI and Scaling the Compute for the new Moore's Law

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 sieć 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](https://pl.wikipedia.org/wiki/Neuron)

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

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:

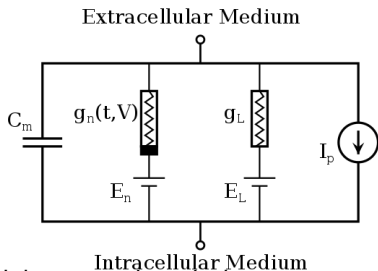
$$a(t) = F(I(t), a(t-1))$$

Aktywacja, razem z progiem wzbudzenia T , określa sygnał wyjściowy

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

Model Hodgkin–Huxley (1952)

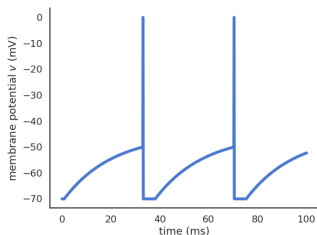
- pierwszy szczegółowy model matematyczny neuronu oparty na przewodnictwie
- nieliniowe równania różniczkowe, które przybliżają charakterystykę elektryczną komórek pobudliwych, takich jak neurony i komórki mięśniowe
- zmiana potencjału błonowego $V(t)$ (między wewnątrz a zewnątrz neuronu) uwzględniająca prądy kanałów jonowych sodowych (K), potasowych (Na), prądy upływu (L) i prąd zewnętrzny $I(t)$ (np. z synaps)



$$C_m \frac{dV(t)}{dt} = -g_K (V(t) - V_K) - g_{Na} (V(t) - V_{Na}) - g_L (V(t) - V_L) - I(t)$$

Neurony impulsujące (*spiking neurons*)

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



$$C_m \frac{dV(t)}{dt} = -g_L (V(t) - V_L) + I(t)$$

Gdy $V > V_T$ (potencjał spoczynkowy) neuron emituje impuls a następnie odpoczywa

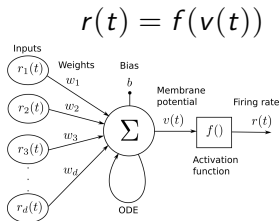
Neurony kodowane częstością wzbudzeń

Rate-coded neurons

- potencjał membrany $V(t)$

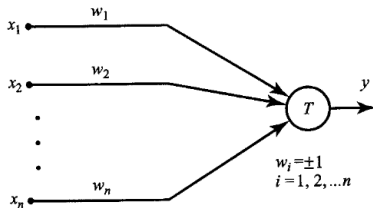
$$\tau \frac{dV(t)}{dt} + V(t) = \sum_{i=1}^d w_{i,j} r_i(t) + b$$

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



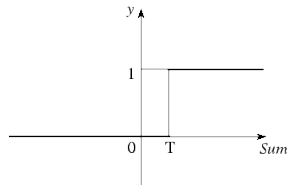
- nie symuluje pojedynczych impulsów (jak np. LIF), ale skupia się na średniej aktywności

Neuron McCulloch, Pitts (1943-49)



stan wyjściowy neuronu

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



grafika: www.wikipedia.org

Neuron McCulloch, Pitts (1943-49)

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

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

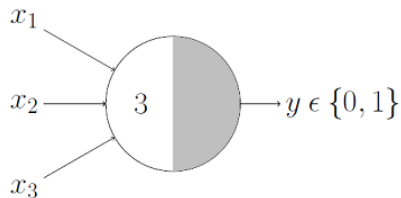
- sygnały dochodzą przez synapsy pobudzające i hamujące

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

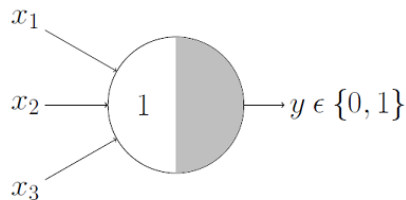
- każdy neuron ma ustalony próg pobudzenia $T > 0$
- sygnały sumują się w pewnym kwancie czasu i gdy przekroczą próg T to neuron się aktywuje
- brak uczenia się - wagi i próg mają stałą wartość
- odpowiedni dobór wag pozwala zrealizować funkcje logiczne: NOT, OR, AND bądź NOR i NAND $\rightarrow$ sieć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

Realizacja funkcji logicznych

AND



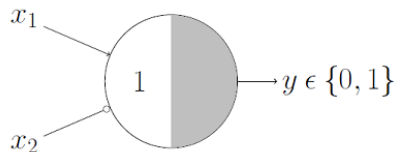
OR



NOT



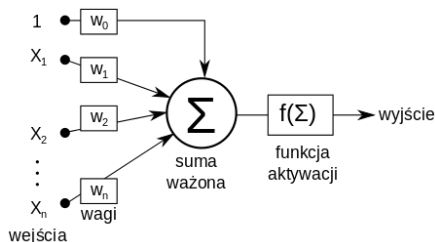
hamowanie sygnału wejściem x_2



Źródło: Akshay L. Chandra,  McCulloch-Pitts Neuron

$$x_1 \text{ AND } !x_2^*$$

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

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

$\mathbf{w} = [w_0, w_1, w_2, \dots, w_n]$ wagi połączeń $\mathbf{w} \in \mathbb{R}^{n+1}$

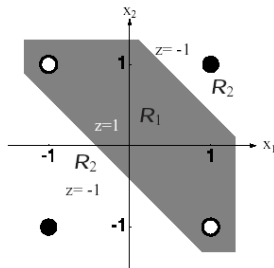
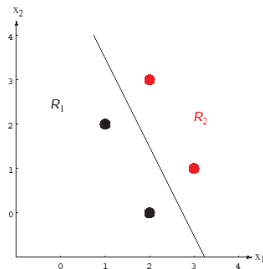
w_0 - wyraz wolny (bias), stały sygnał wejściowy $1 \cdot w_0$

Interpretacja geometryczna - separowalność

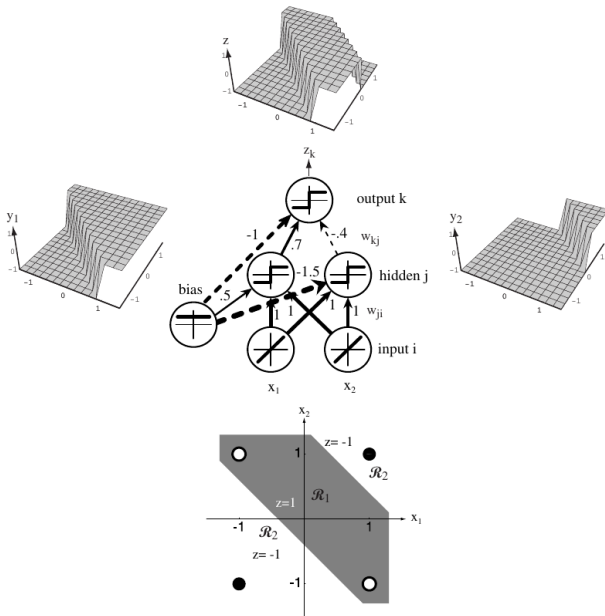
Neuron z progiem realizuje separowalność liniową, hiperpłaszczyzna dzieli przestrzeń wejściową $\mathbb{R}^n$ na dwie części

Przykład neuronu z 2 wejściami

$$o(\mathbf{x}) = \begin{cases} 1 & \text{gdy } w_1x_1 + w_2x_2 \geq \theta \\ 0 & \text{gdy } w_1x_1 + w_2x_2 < \theta \end{cases}$$



XOR



Reguła Hebba (1949)

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

„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

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

Reguła delta (Widrowa-Hoffa)

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

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

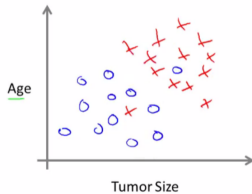
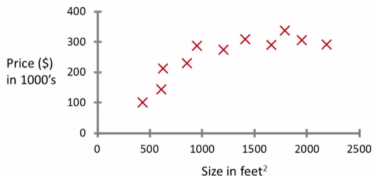
Rodzaje ucznia

- **Uczenie bez nadzoru** (*unsupervised learning*)
uczenie wyłącznie na podstawie sygnału wejściowego $\mathbf{X}$ (bez etykiet $\mathbf{y}$), odkrywanie struktur i zależności w danych.
Uczenie spontaniczne, odpowiada uczeniu w okresie niemowlęcym.
- **Uczenie nadzorowane** (*supervised learning*) - dla każdego sygnału wejściowego $\mathbf{x}$ dana jest pożądana odpowiedź $\mathbf{y}$.
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.:

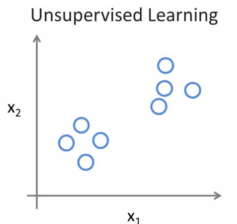
$$(f(\mathbf{x}; \mathbf{W}) - \mathbf{y})^2$$

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

Uczenie nienadzorowane

Uczenie wyłącznie na podstawie sygnału wejściowego $\mathbf{X}$, brak pożądanego sygnału wyjściowego. Przykłady:

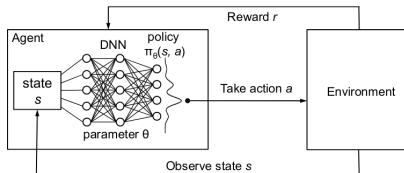
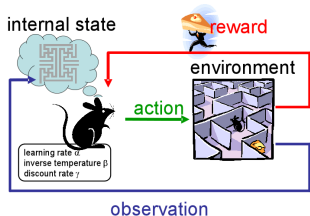
- **klasteryzacja** (analiza skupień), automatyczne wykrywanie grup o podobnych właściwościach



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

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)

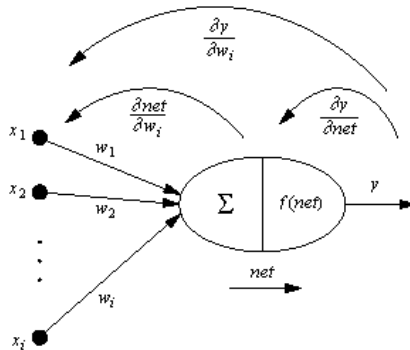


Rysunek: Hongzi Mao, „Resource Management with Deep Reinforcement Learning”
Aneek Das, The very basics of Reinforcement Learning

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), ...



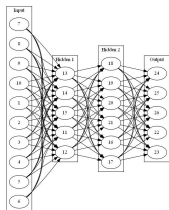
Główne aspekty modeli neuronowych II

2. Sposób propagacji sygnałów przez sieć neuronową

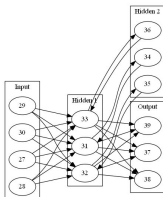
- sieci jednokierunkowe (feedforward)
- sieci ze sprzężeniami zwrotnymi, sieci rekurencyjne (recurrent), sieci dynamiczne

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, współdzielenie połączeń, split
- sieci o stałej architekturze, sieci ontogeniczne (zmieniające strukturę), rozrastające się, kurczące się,

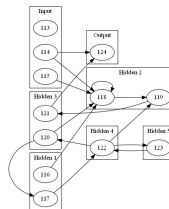


Neural Network



Neural Network

Sieci Neuronowe



Neural Network

Główne aspekty modeli neuronowych III

4. Uczenie sieci
 - reguły uczenia, reguły modyfikacji parametrów opisujących neurony i połączeń 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