



24.03.2023

SZTUCZNE SIECI NEURONOWE

Małgorzata Trąbka

Plan warsztatu

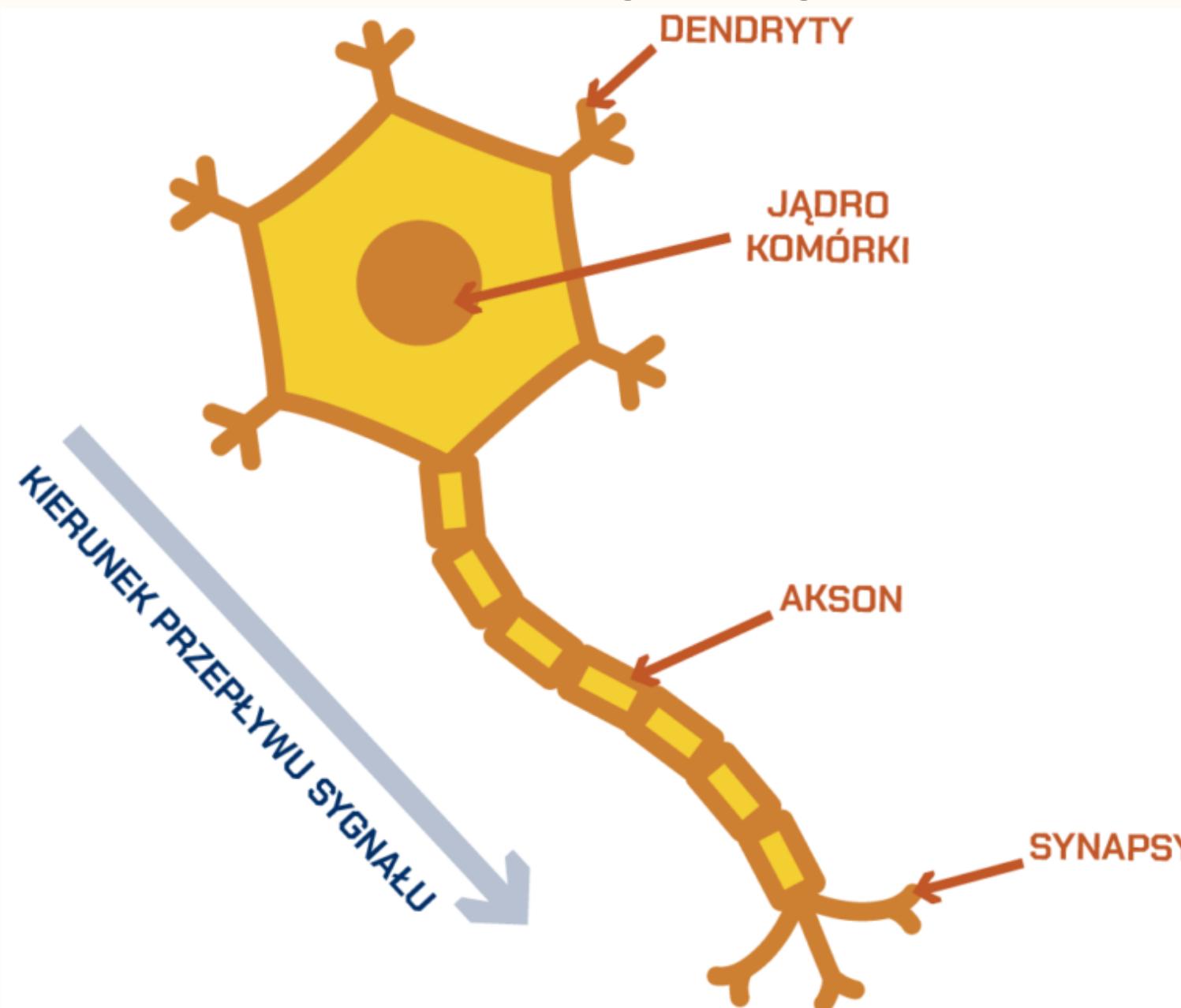
- 1 Czym są sztuczne sieci neuronowe?
- 2 Budowa sztucznych sieci neuronowych
- 3 Rodzaje sieci neuronowych
- 4 Tensorflow Playground

Sztuczne sieci neuronowe

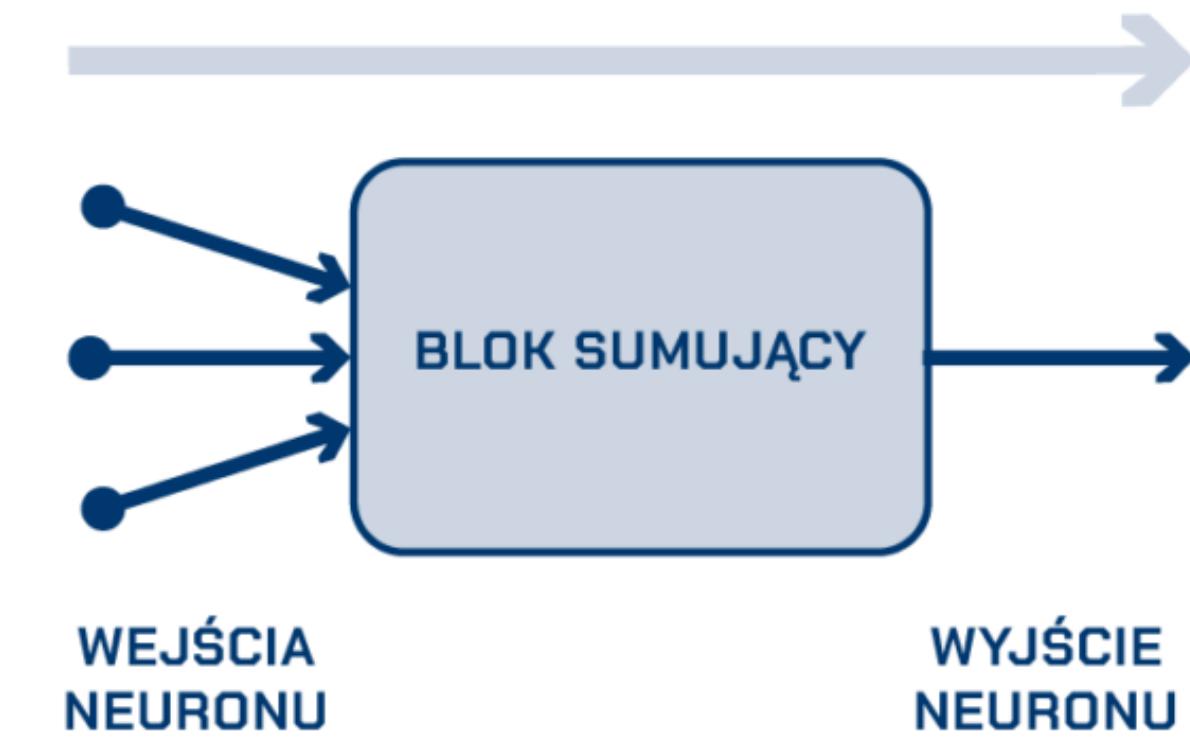
Sztuczne sieci neuronowe to model matematyczny, który naśladuje sposób działania mózgu, w którym wiele prostych jednostek, tzw. neuronów, łączy się ze sobą, aby wykonywać bardziej skomplikowane obliczenia.

Neurony

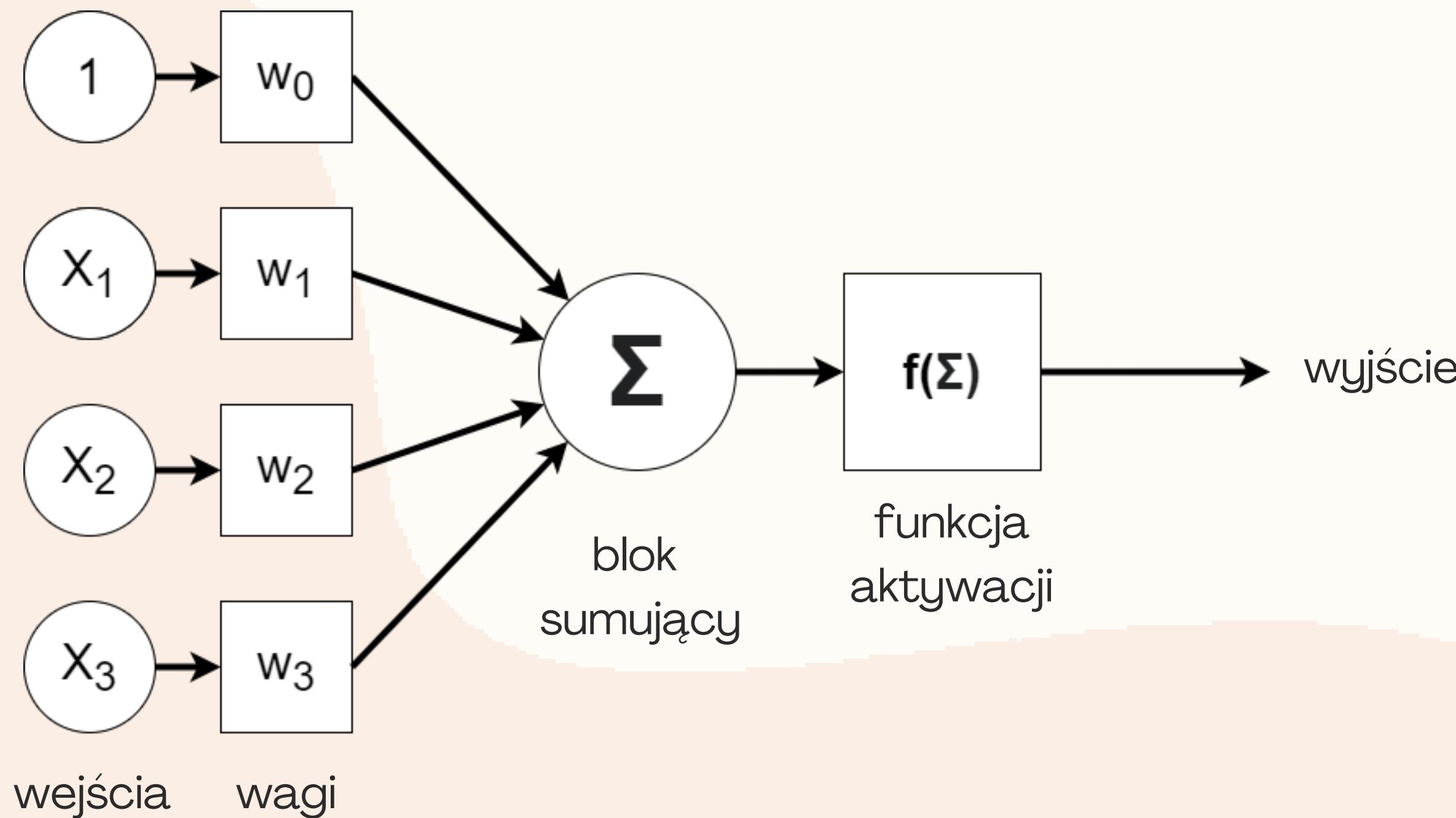
Budowa neuronu biologicznego



Najprostszy model sztucznego neuronu

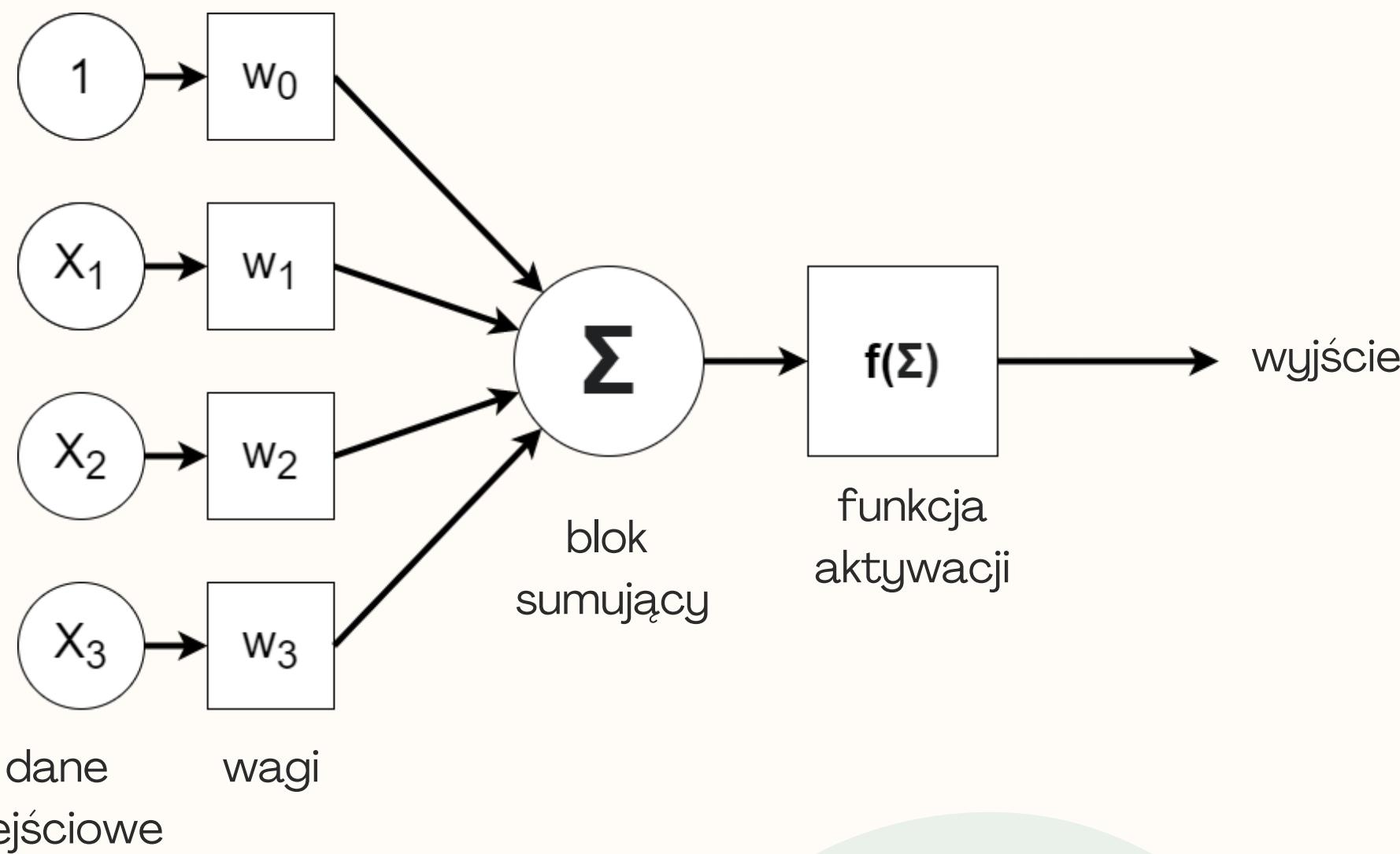


Jak działa sztuczny neuron?



Dane wejściowe
Informacje podawane do wejścia neuronu

Wagi
Parametr decydujący o „ważności” danego wejścia neuronu



Blok sumujący
Blok przekształcający wiele danych wejściowych w jeden sygnał

Funkcja aktywacji
Określa sposób obliczania wartości sygnału wyjściowego neuronu

$$S = \sum_{i=0}^n x_i w_i$$

Lista zakupów

Pomidory 1,5 kg

Papryka 2 kg

Marchew 0,8 kg

23,99/kg



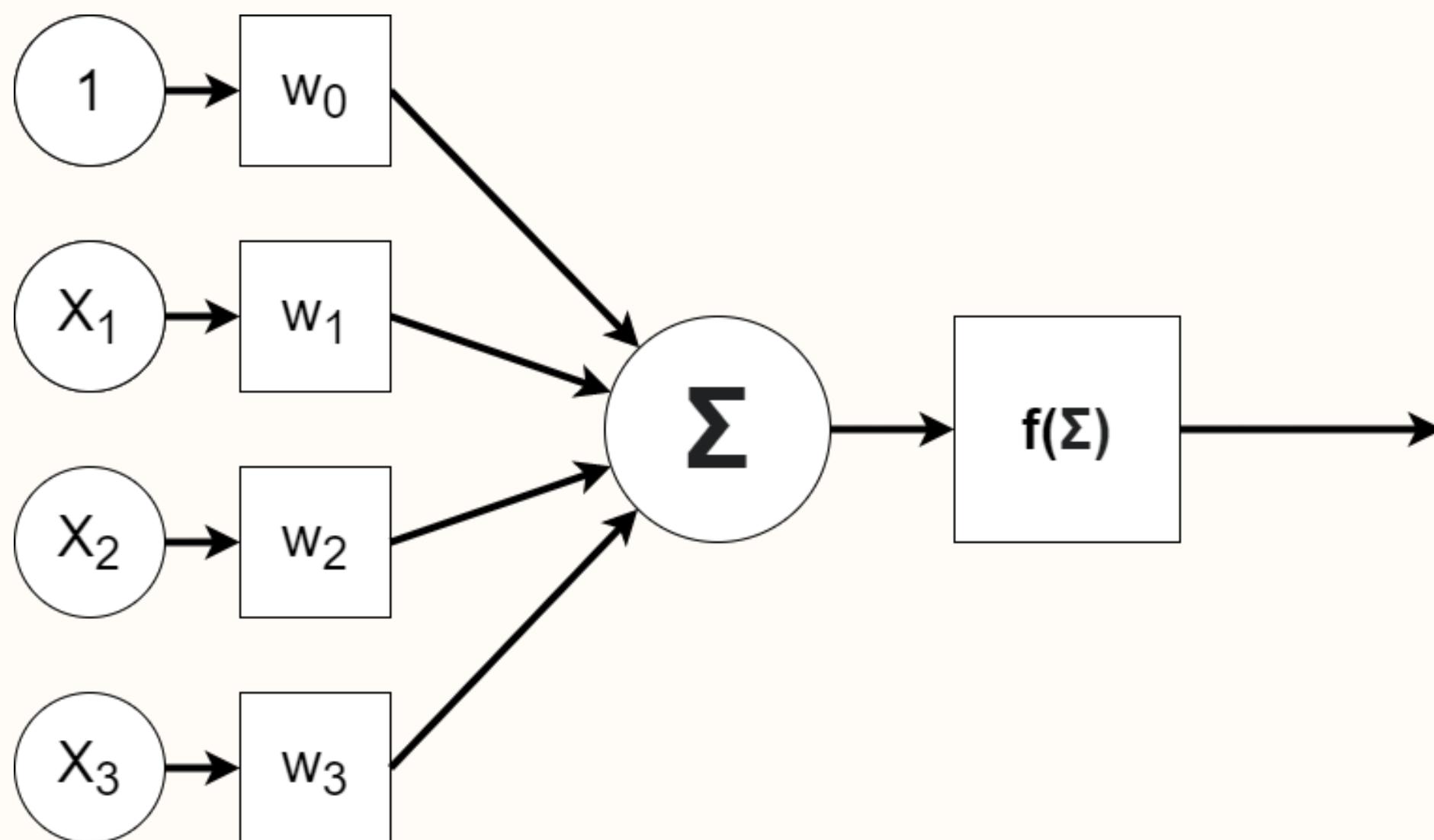
11,99/kg



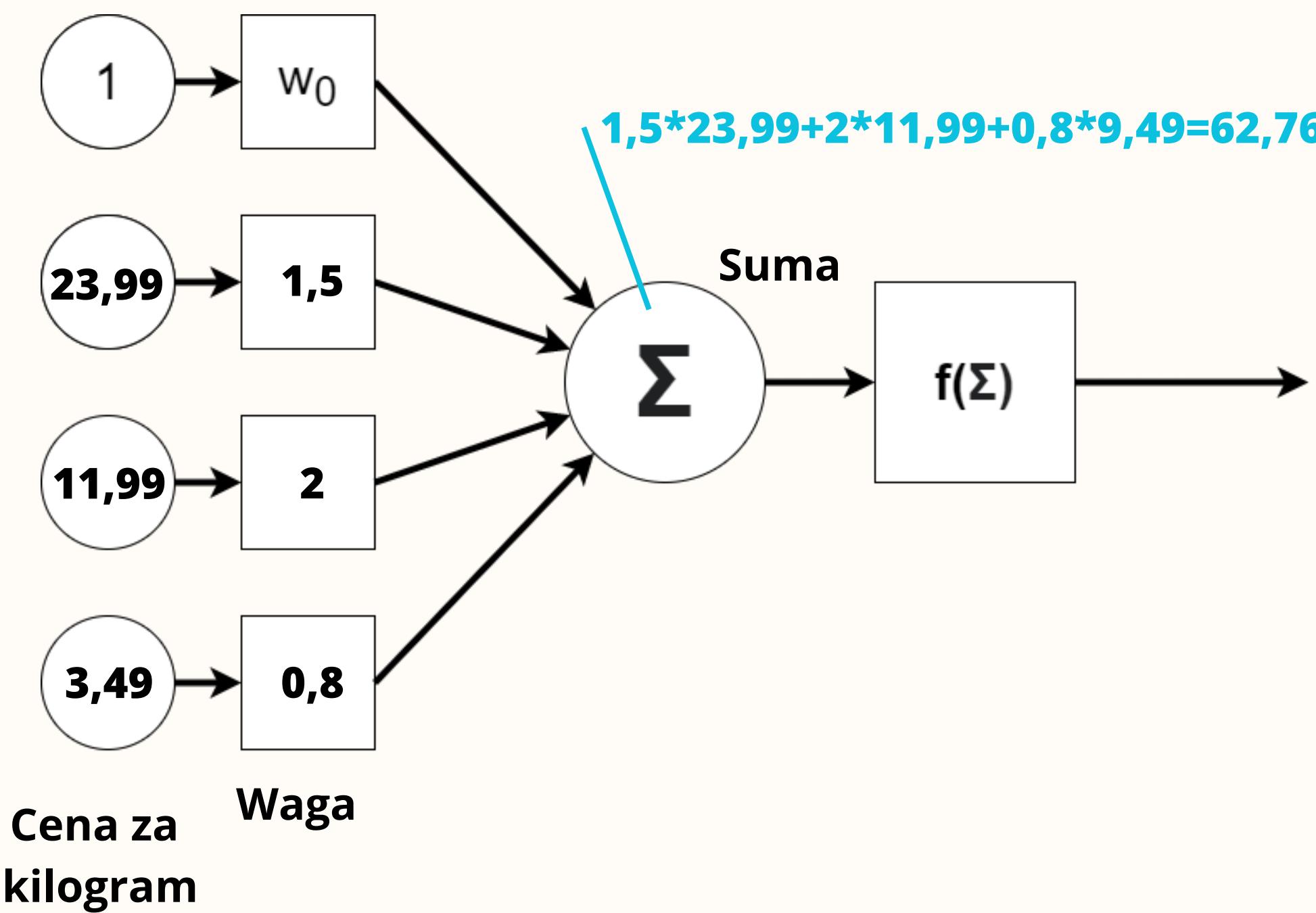
3,49/kg



Nazwa	Waga [kg]	Cena za kilogram [zł]	Cena [zł]
Pomidory	1,5	23,99	35,99
Papryka	2	11,99	23,98
Marchew	0,8	3,49	2,79
Suma			62,76



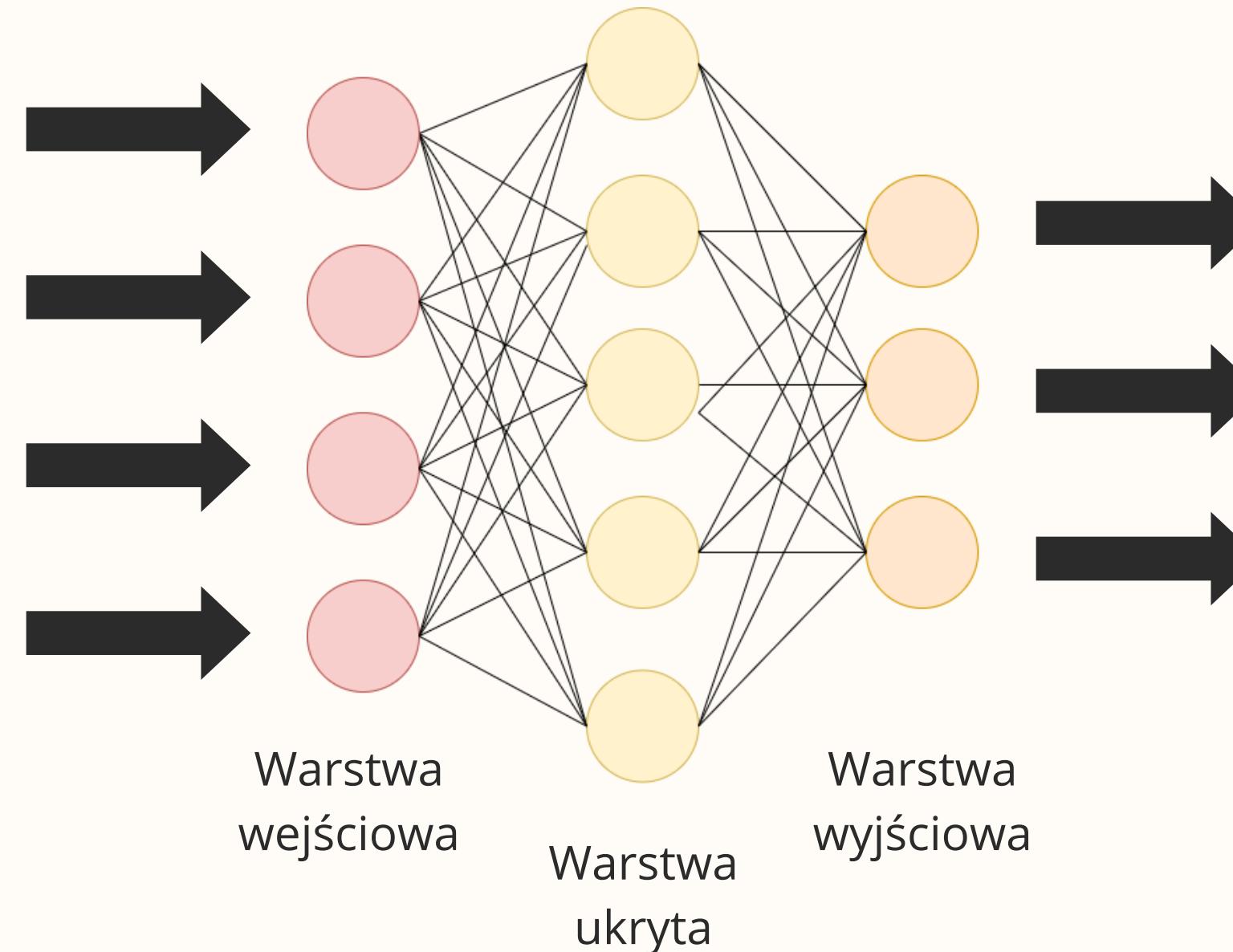
Nazwa	Waga [kg]	Cena za kilogram [zł]	Cena [zł]
Pomidory	1,5	23,99	35,99
Papryka	2	11,99	23,98
Marchew	0,8	3,49	2,79
Suma			62,76



Sztuczna sieć neuronowa

Sieć neuronowa to wiele połączonych ze sobą sztucznych neuronów

Warstwa wejściowa – zbiera dane i przesyła je dalej (każdy neuron z warstwy wejściowej przesyła dane do każdego neuronu w warstwie ukrytej)

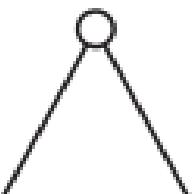
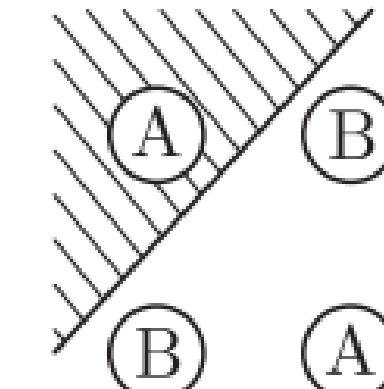
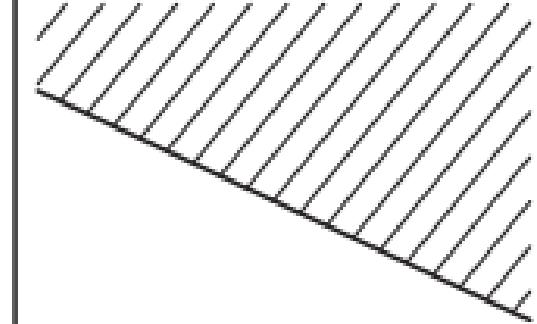
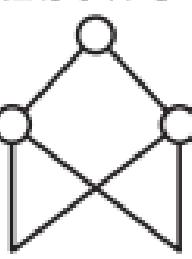
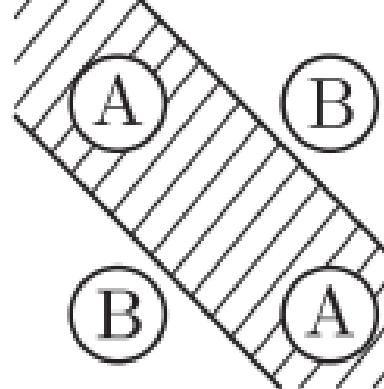
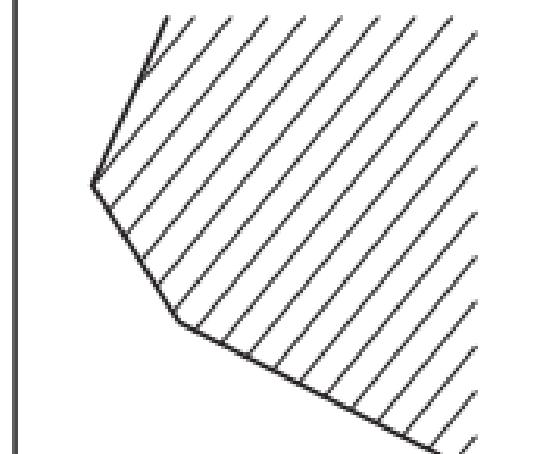
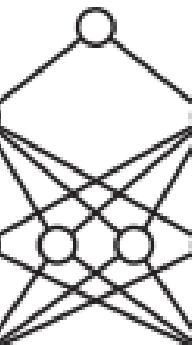
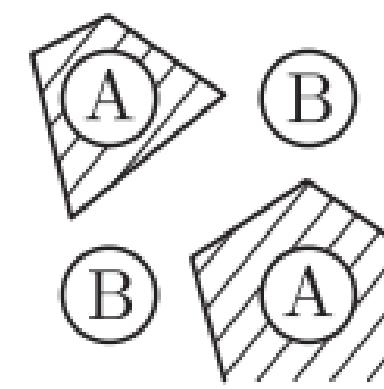
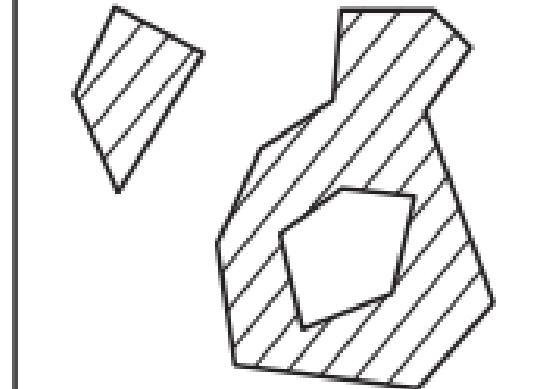


Warstwa wyjściowa – zwraca wynik

Warstwa ukryta – są to stany pośrednie. Tutaj przede wszystkim zachodzi proces uczenia się i szukana liniowych i nieliniowych zależności. Może być wiele warstw ukrytych.

Struktura sztucznej sieci neuronowej

- **liczba warstw ukrytych (głębokość sieci)** – za mało warstw prowadzi do niedotrenowania sieci, za dużo – do przetrenowania
- **liczba neuronów w poszczególnych warstwach** – źle dobrana ilość może skutkować przetrenowaniem lub niedotrenowaniem sieci. Liczba neuronów wejściowych może odzwierciedlać ilość cech danych wejściowych, a liczba neuronów warstwy wyjściowej powinna być dopasowana do typu problemu:
 - regresja – jeden
 - klasifikacja binarna – jeden
 - klasifikacja wieloklasowa – tyle ile jest rozłącznych klas

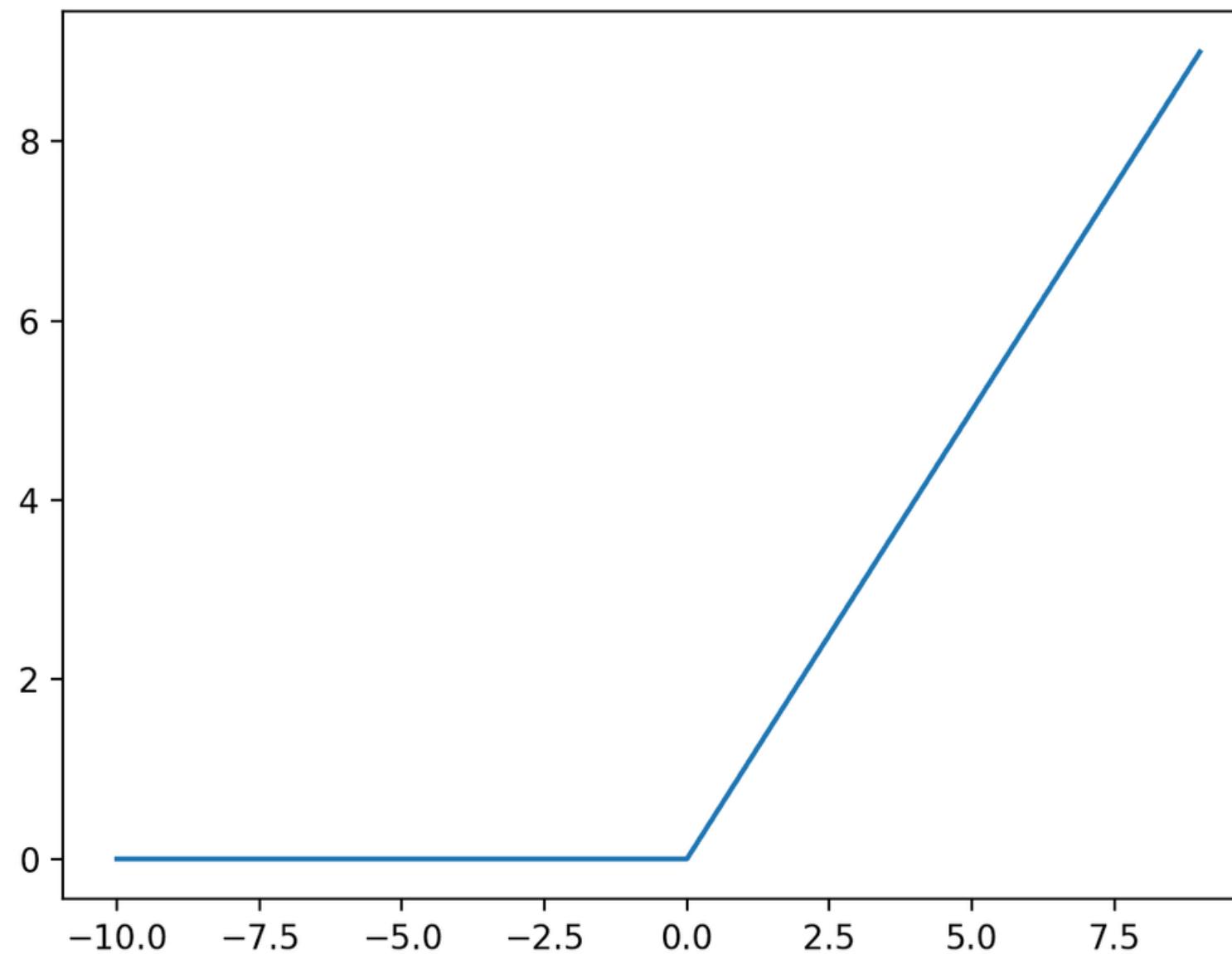
Struktura	Typ obszarów decyzyjnych	Problem XOR	Najbardziej ogólne kształty obszarów
Jedno-warstwowa 	Półpłaszczyzna ograniczona przez hiperpłaszczyznę		
Dwu-warstwowa 	Wypukłe obszary otwarte lub zamknięte		
Trój-warstwowa 	Dowolne (złożoność ograniczona liczbą neuronów)		

Zródło: <https://dominicm73.blogspot.com/2020/08/modeling-threshold-logic-neurons-and.html>

Struktura sztucznej sieci neuronowej

- **Funkcja aktywacji** - funkcja matematyczna, która jest zastosowana w sieciach neuronowych, w celu wprowadzenia nieliniowości do modelu. Bez funkcji aktywacji, wynik działania każdej warstwy byłby tylko kombinacją liniową wag wejściowych i biasów.
- **Batch size** - liczba danych uczących, które są przetwarzane jednocześnie w jednej iteracji podczas trenowania sieci neuronowej
- **Epoch/Epoka** - przejście przez cały zbiór danych uczących
- **Learning rate** - hiperparametr sieci neuronowej, który określa jak szybko sieć ma się uczyć poprzez modyfikację wag.
- **Regularyzacja** - technika, która ma na celu zapobieganie overfittingu, czyli zbytniemu dopasowaniu modelu do danych uczących.

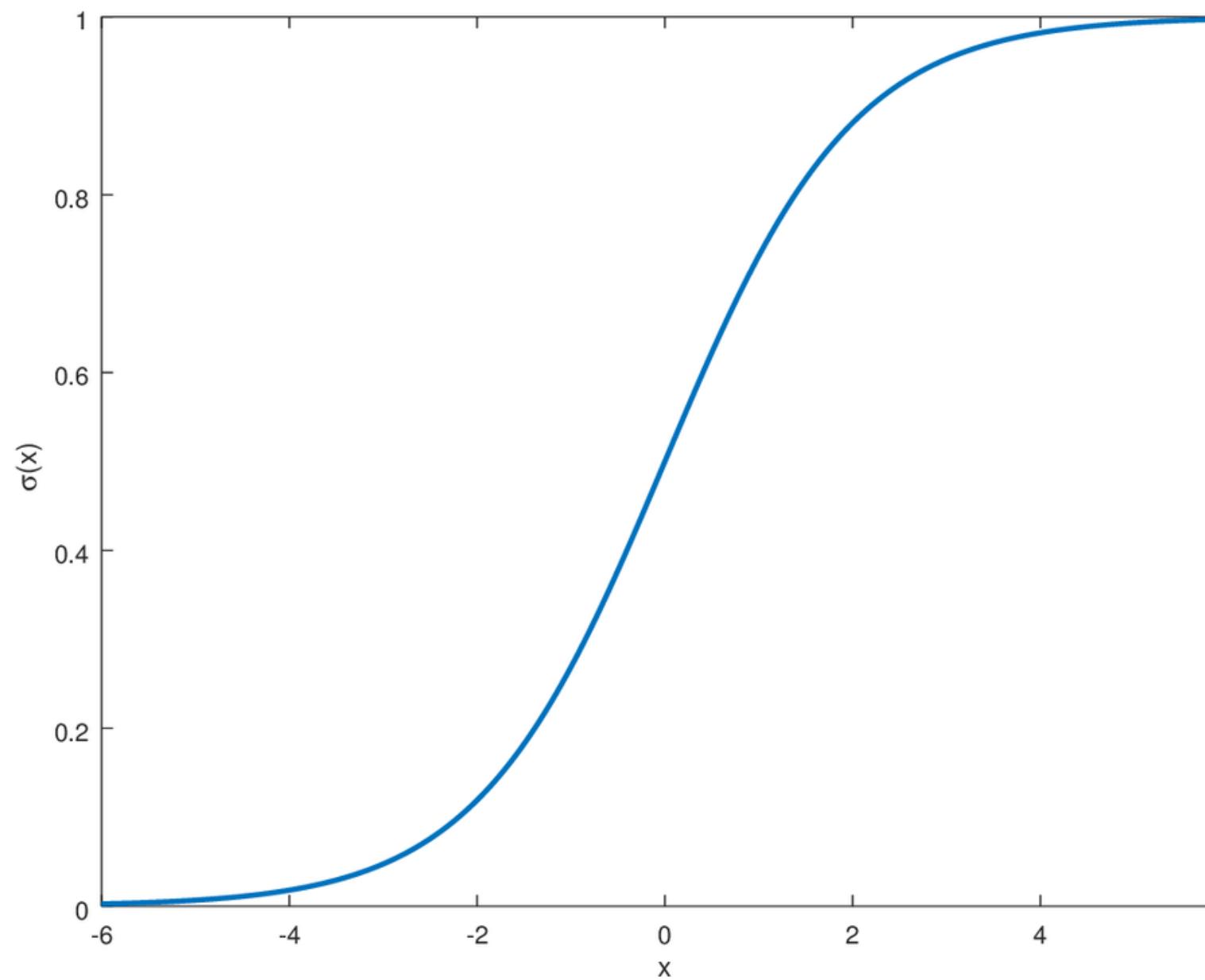
Funkcje aktywacji - ReLU



$$f(x) = x^+ = \max(0, x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

- Uważa się, że zbieżność ReLU jest 6 razy szybsza niż funkcji sigmoid i tanh.
- Domyślnie używa się jej w sieciach MLP (multilayer perceptron) i CNN (convolutional neural network).
- Nigdy nie używa się jej w warstwie wyjściowej.

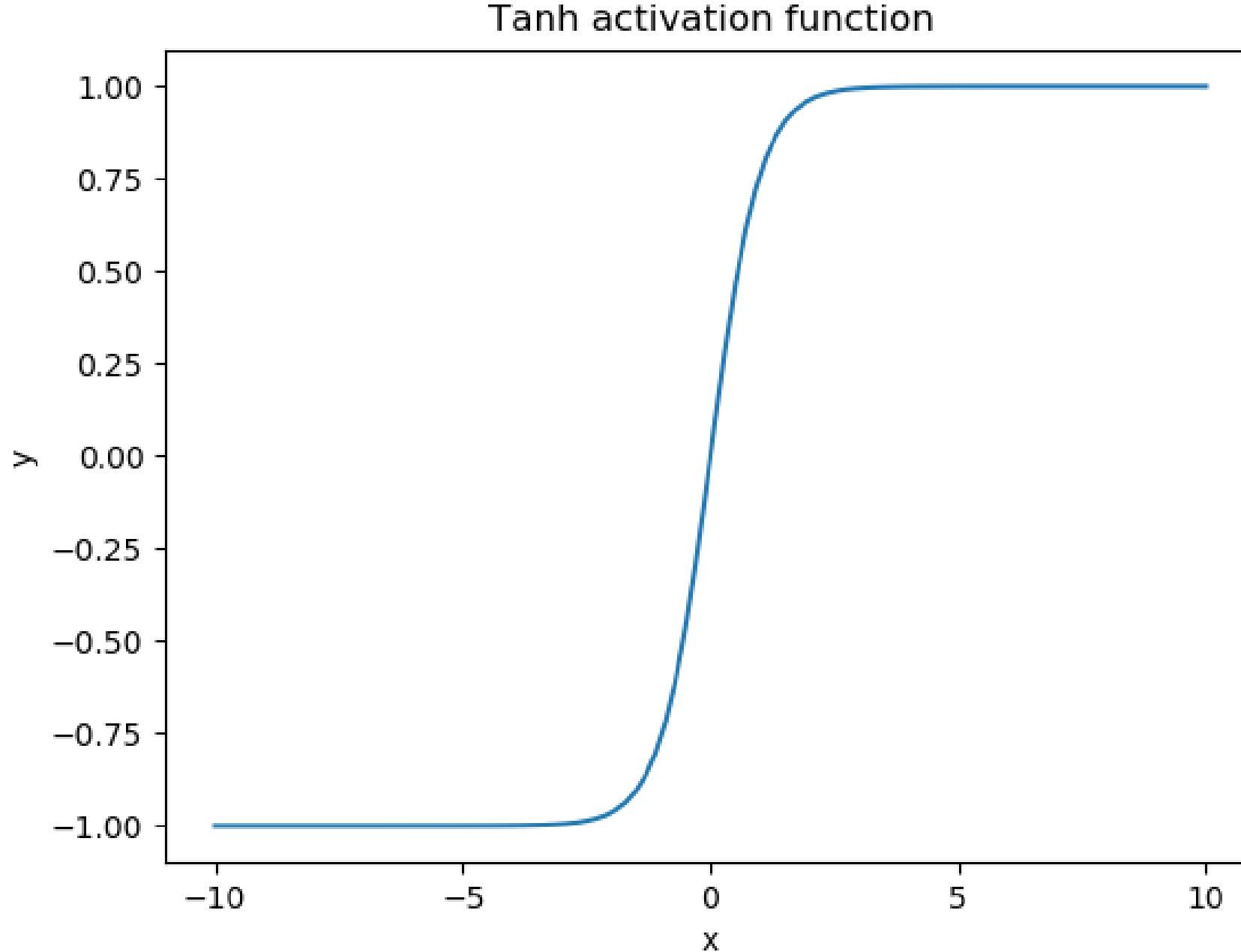
Funkcje aktywacji – Sigmoid



$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

- Domyślnie używa się jej w sieciach RNN (recurrent neural network).

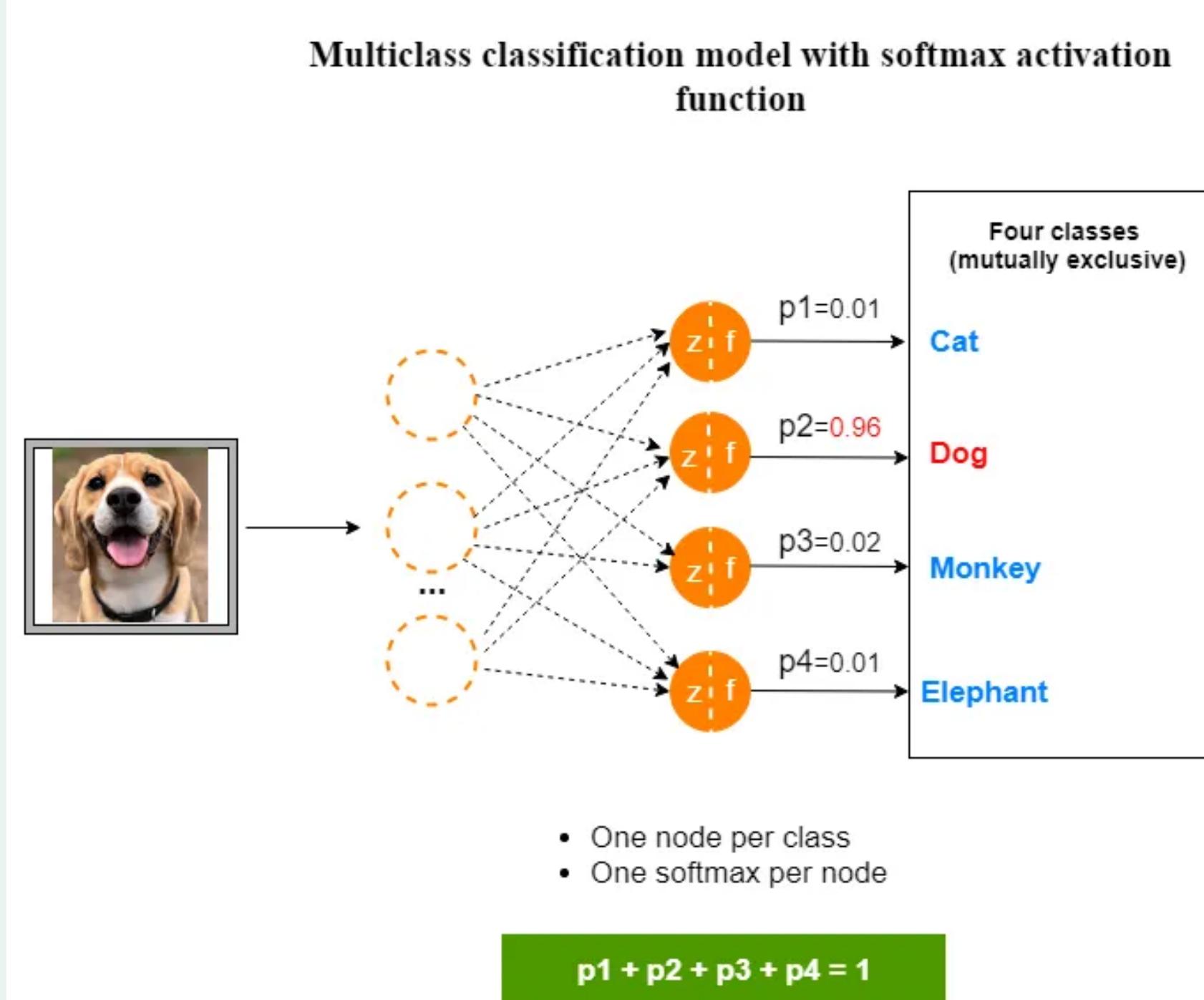
Funkcje aktywacji - tanh



$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

- Przyjmuje wartości od -1 do 1.
- Używana domyślnie w sieciach RNN (recurrent neural network).
- Nigdy nie używa się jej w warstwie wyjściowej.

Funkcje aktywacji – softmax



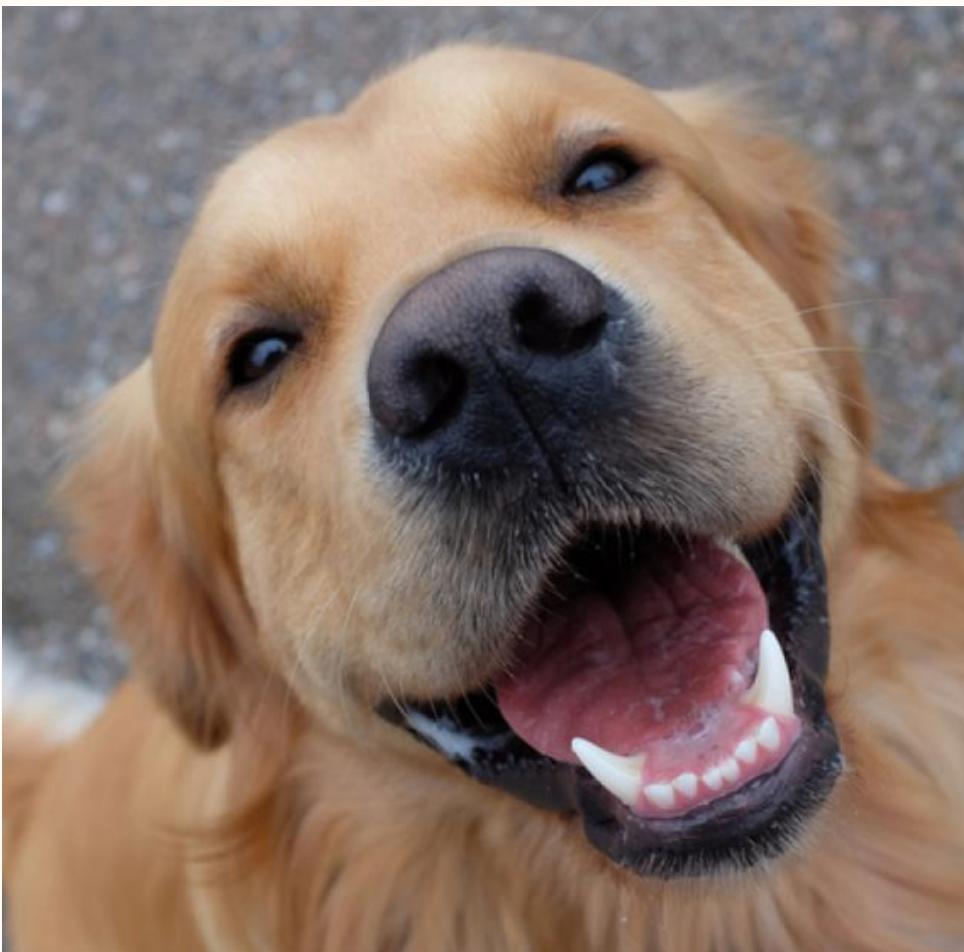
$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

- Oblicza wartość prawdopodobieństwa zdarzenia klasy
- Suma wszystkich prawdopodobieństw wynosi 1 – klasy wzajemnie się wykluczają
- Używa się jej w warstwie wyjściowej przy klasyfikacji wieloklasowej
- Nie używa się jej w warstwach ukrytych

Konwolucyjne sieci neuronowe

- Sieci konwolucyjne wykorzystują operację konwolucji do wyodrębniania cech z obrazów, dzięki czemu są w stanie rozpoznawać wzorce w obrazach i dokonywać klasyfikacji.
- W sieciach konwolucyjnych dane wejściowe są przetwarzane przez serię warstw, w których wykorzystywane są filtry konwolucyjne. Filtry te przesuwane są po obrazie i wykonywana jest operacja konwolucji, która pozwala na wydobycie ważnych cech, takich jak krawędzie, linie czy kształty.
- Wynik konwolucji jest przekazywany do warstw poolingowych, które zmniejszają wymiary danych, co pozwala na szybsze przetwarzanie.
- Ostatnią warstwą w sieciach konwolucyjnych jest zwykle warstwa w pełni połączona, która dokonuje klasyfikacji na podstawie wyodrębnionych cech.
- Sieci konwolucyjne są wykorzystywane w wielu dziedzinach, takich jak rozpoznawanie obrazów, rozpoznawanie mowy, czy analiza sygnałów.

Detekcja psich emocji

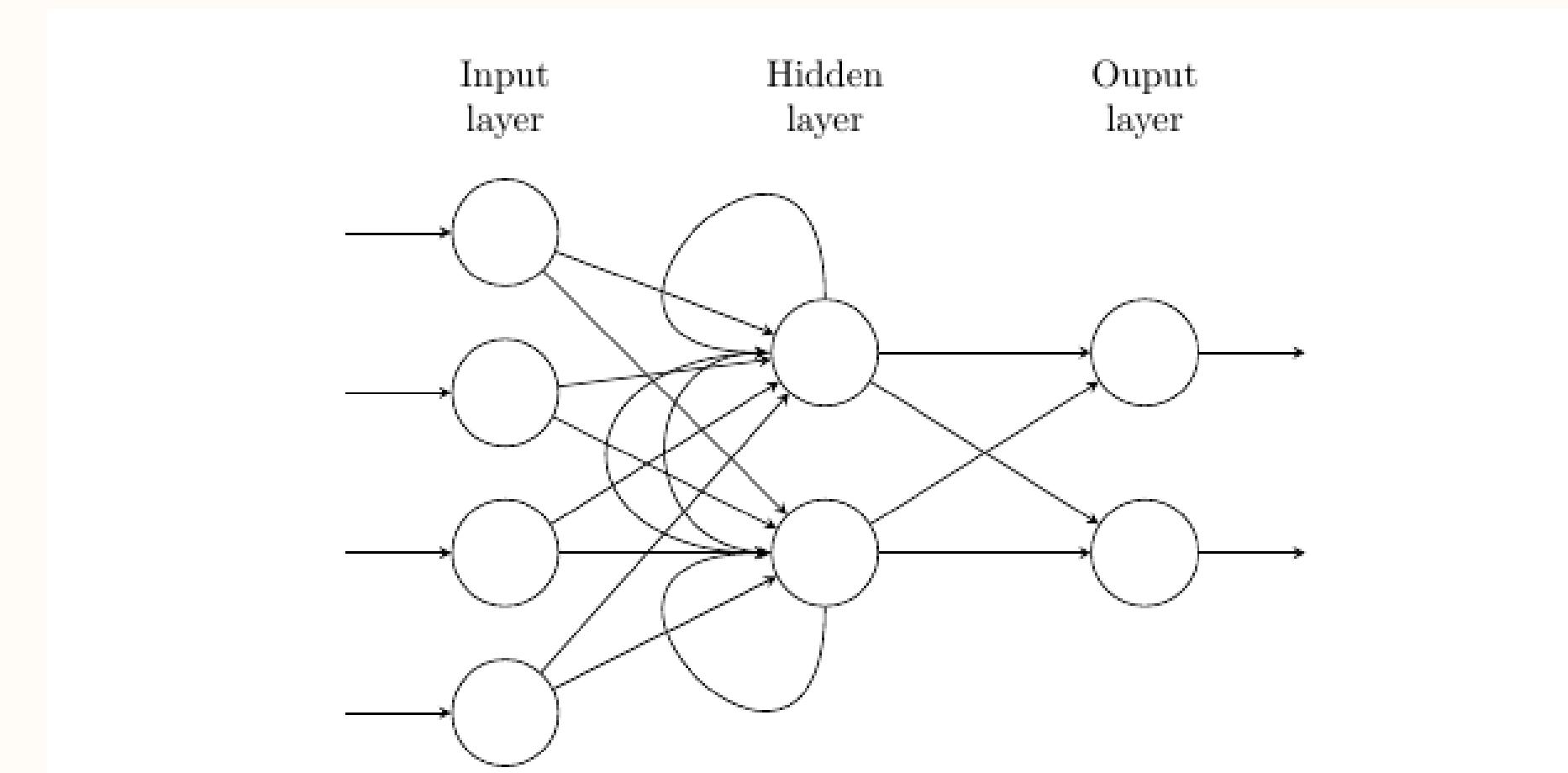


Sieć
konwolucyjna

Happy
Angry
Sad
Relaxed

Sieć rekurencyjna (RNN)

- Struktura zaprojektowana dla danych sekwencyjnych.
- Dane sekwencyjne to wszelkie dane, które występują w formie, w której wcześniejsze punkty danych wpływają na późniejsze punkty danych.
- Sieci RNN są rozszerzane w stosunku do podstawowej sieci neuronowej typu feed-forward przez dodanie pętli sprzężenia zwrotnego dla co najmniej jednej warstwy neuronów.



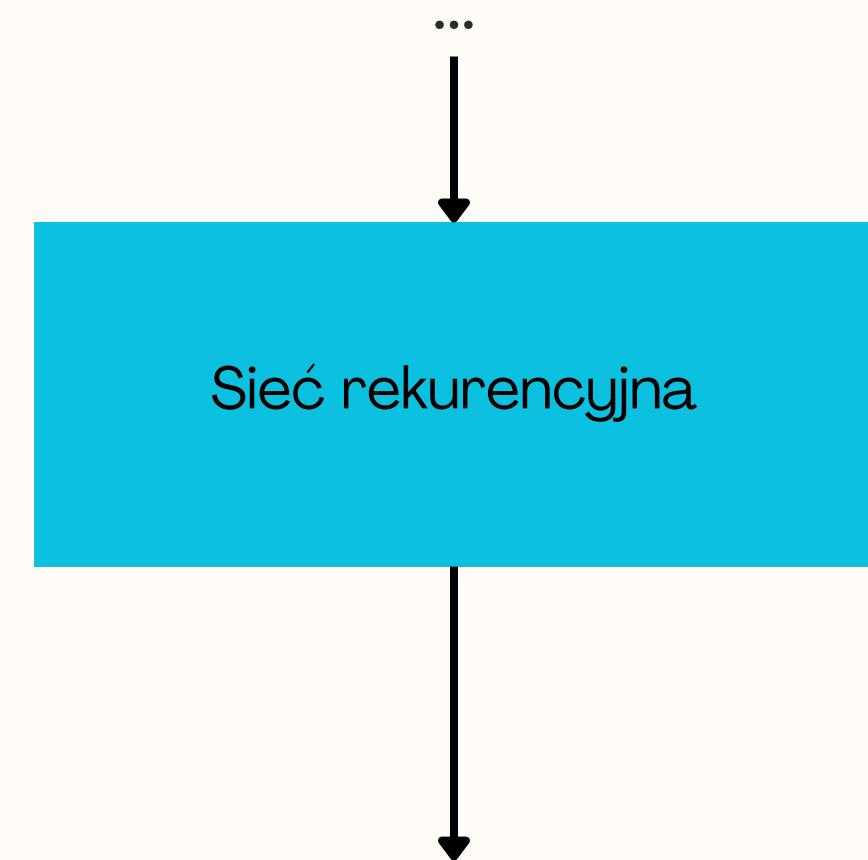
Generator Hate Speech

„Ukainka zrobiła z Pazury wała”

„Odezwał się fan Rydzyka też chory z nienawiści do Żydów”

„Drogi Panie Tomaszu coby nie zabrzmieć jak gej ale jest pan przystojny i elegancki”

Źródło: https://huggingface.co/datasets/hate_speech_pl

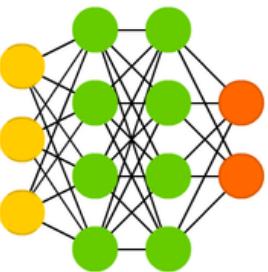


A mostly complete chart of
Neural Networks

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool

©2016 Fjodor van Veen - asimovinstitute.org

Deep Feed Forward (DFF)



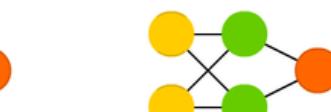
Perceptron (P)



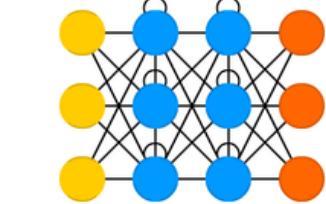
Feed Forward (FF)



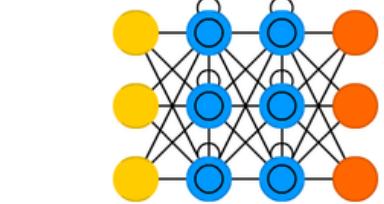
Radial Basis Network (RBF)



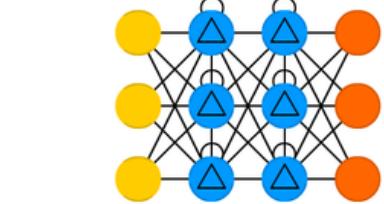
Recurrent Neural Network (RNN)



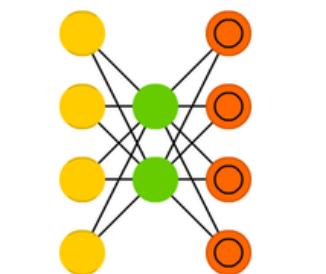
Long / Short Term Memory (LSTM)



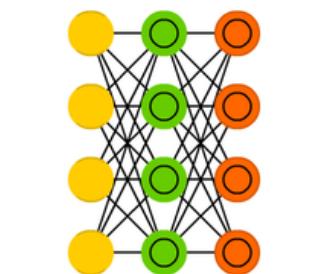
Gated Recurrent Unit (GRU)



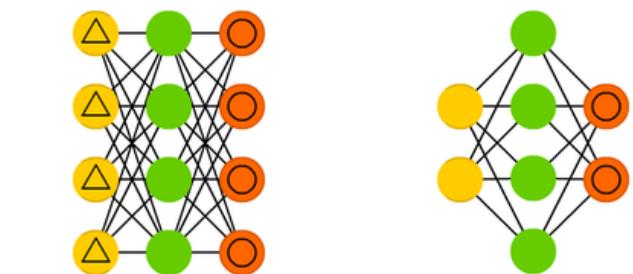
Auto Encoder (AE)



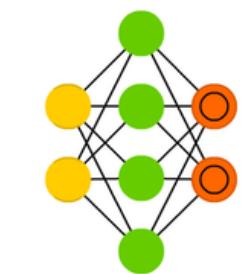
Variational AE (VAE)



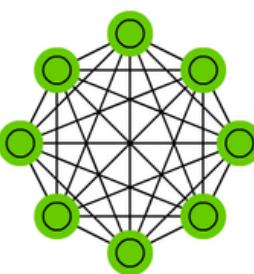
Denoising AE (DAE)



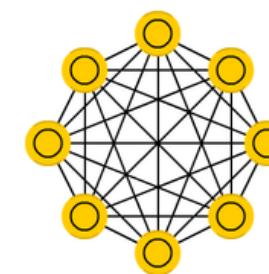
Sparse AE (SAE)



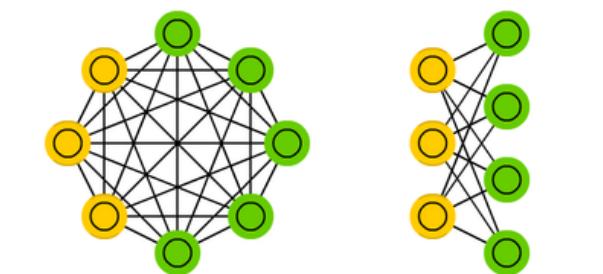
Markov Chain (MC)



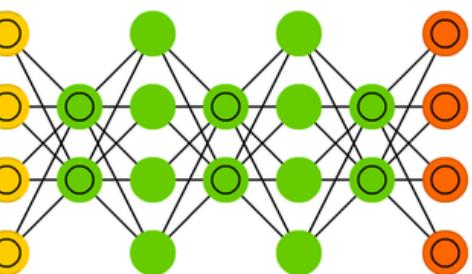
Hopfield Network (HN)



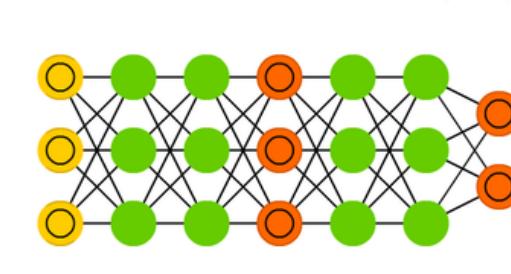
Boltzmann Machine (BM) Restricted BM (RBM)



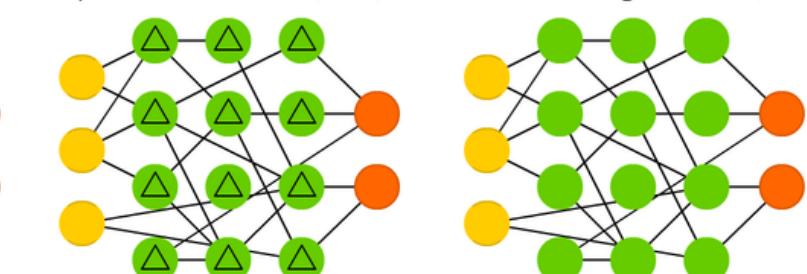
Deep Belief Network (DBN)



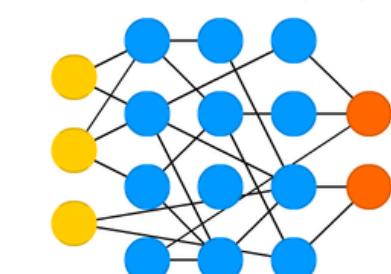
Generative Adversarial Network (GAN)



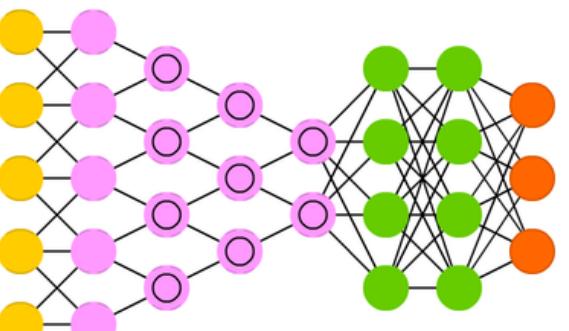
Liquid State Machine (LSM) Extreme Learning Machine (ELM)



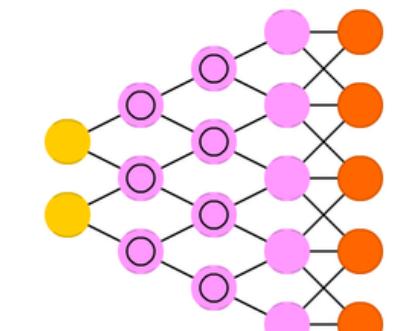
Echo State Network (ESN)



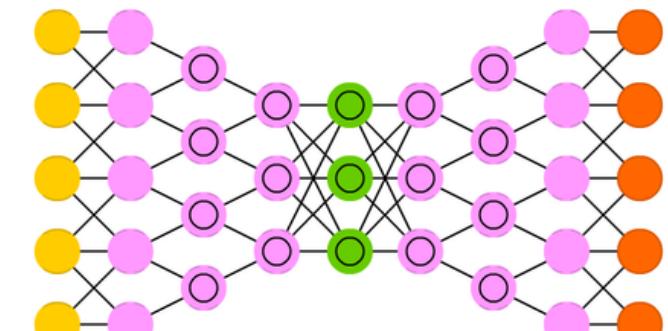
Deep Convolutional Network (DCN)



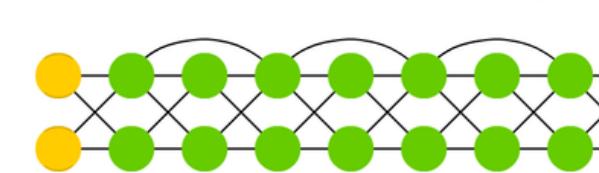
Deconvolutional Network (DN)



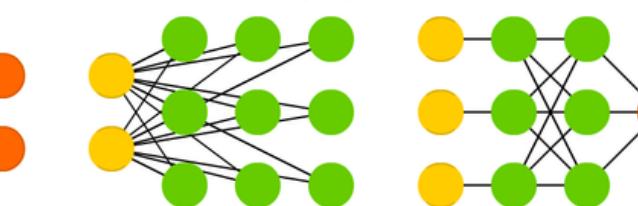
Deep Convolutional Inverse Graphics Network (DCIGN)



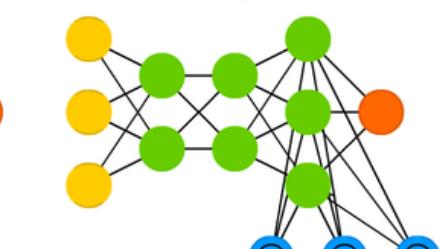
Deep Residual Network (DRN)



Kohonen Network (KN) Support Vector Machine (SVM)



Neural Turing Machine (NTM)



Różne architektury sieci

<https://playground.tensorflow.org>

Dziękuję za uwagę!

Prezentacja i kody dotychczasowych projektów:

<https://github.com/MTrabka/ssn>

Więcej ćwiczeń w Tensorflow Playground:

<https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises?hl=pl>