



Group of
Horribly
Optimistic
Statisticians



CV SEMINAR

WSTĘP DO GŁĘBOKIEGO UCZENIA 2

28.11.2023 Computer Vision Seminar 23/24



GHOST

Group of Horribly Optimistic Statisticians



Agenda

1. Jeszcze raz o KNN
2. Klasyfikator liniowy
3. Neuron
4. Czym jest głębokie uczenie?
5. Propagacja wsteczna
6. Funkcja straty

Szeliski rozdział 5: Deep Neural Networks

CS231n Module 1: <https://cs231n.github.io/>





GHOST

Group of Horribly Optimistic Statisticians



Wyzwania w klasyfikacja zdjęć

Viewpoint variation



Scale variation



Deformation



Occlusion



Illumination conditions



Background clutter



Intra-class variation



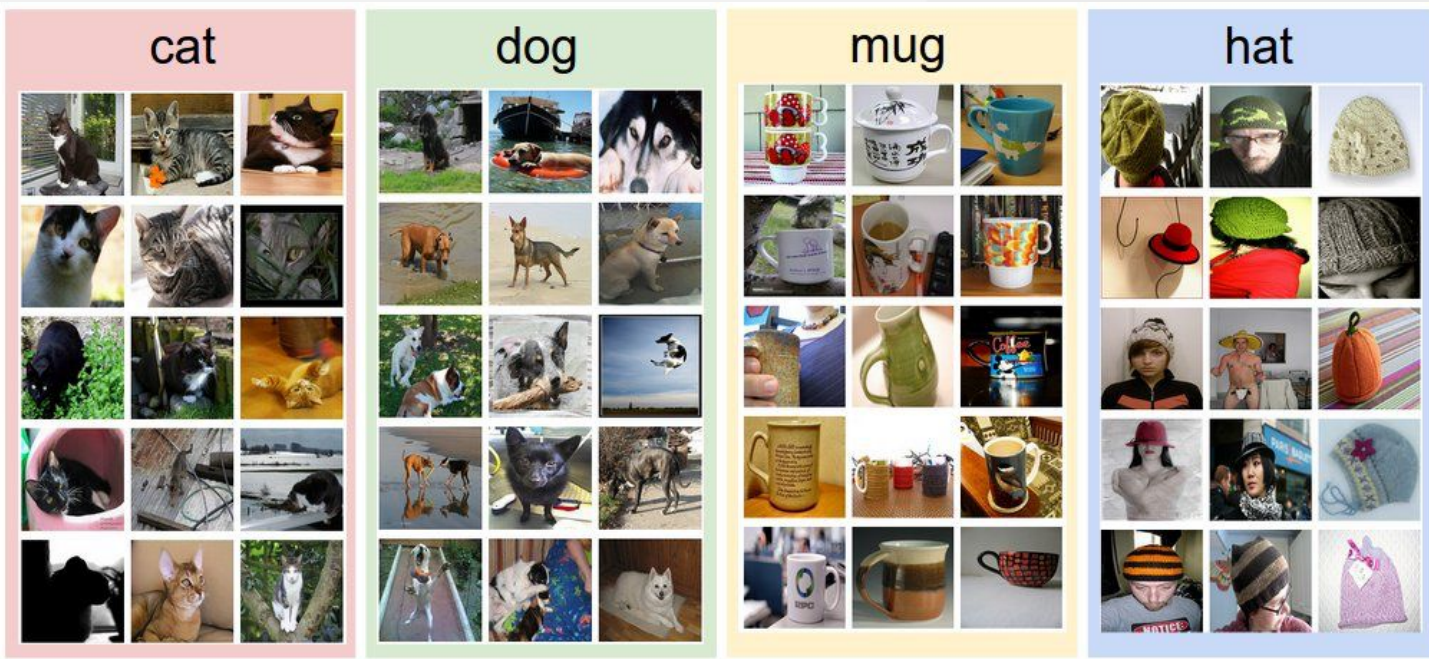


GHOST

Group of Horribly Optimistic Statisticians



Data-driven approach





GHOST

Group of Horribly Optimistic Statisticians



CIFAR-10

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck





GHOST

Group of Horribly Optimistic Statisticians



Dystans per-piksel

original



shifted



messed up



darkened



Pixel-based distances on high-dimensional data (and images especially) can be very unintuitive. An original image (left) and three other images next to it that are all equally far away from it based on L2 pixel distance. Clearly, the pixel-wise distance does not correspond at all to perceptual or semantic similarity.

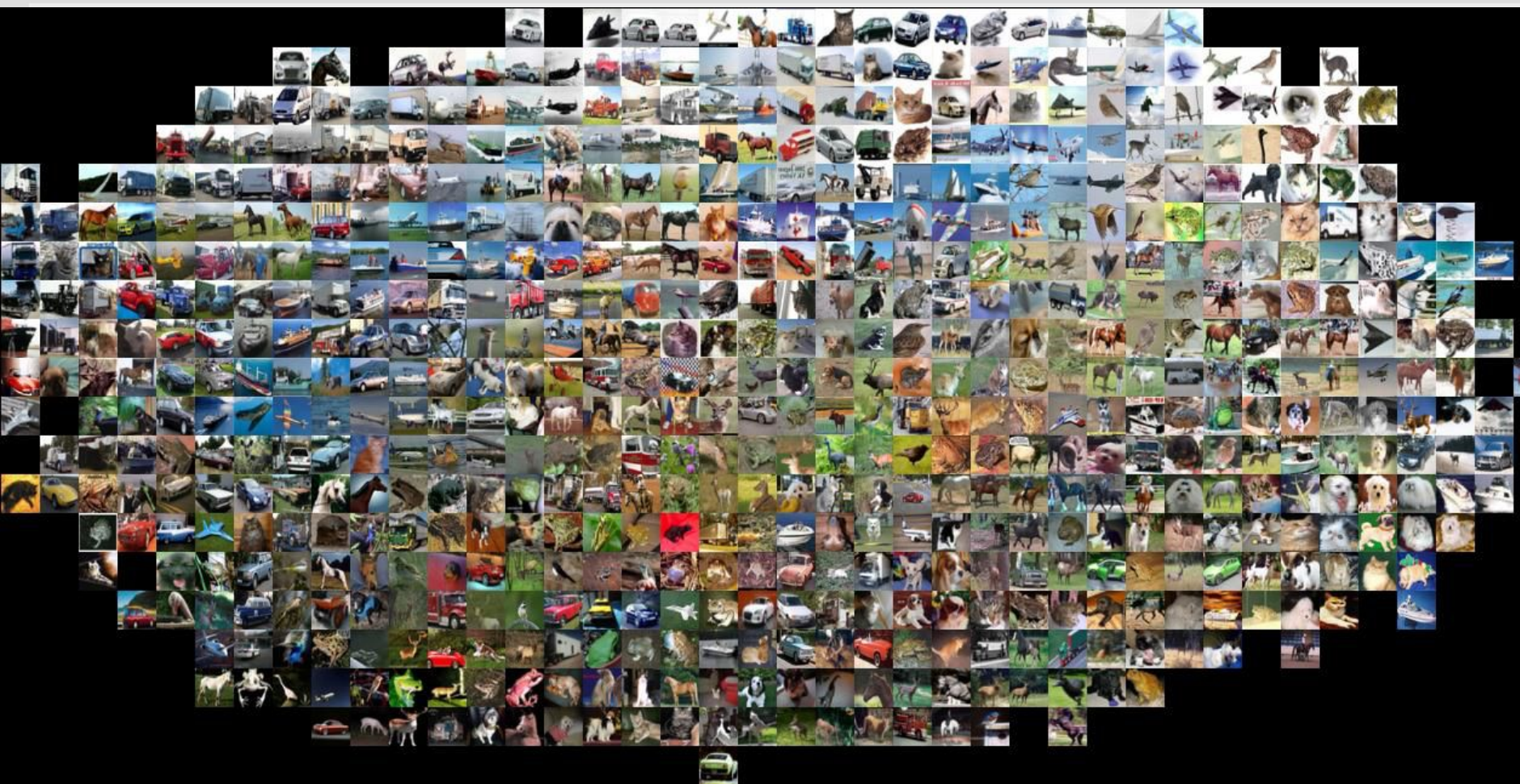


GHOST

Group of Horribly Optimistic Statisticians

Zdjęcie jako wektor w wielu wymiarach







GHOST

Group of Horribly Optimistic Statisticians



Algorytm K najbliższych sąsiadów

Zalety

- Intuicyjny algorytm
- Nie wymaga trenowania

Wady

- **Długi czas inferencji**
- **Wymaga przechowywania wszystkich danych w pamięci**
- Dystanse dla danych o wielu wymiarach nie są intuicyjne
- Dystanse dobrze ujmują kolor i tło, ale nie są w stanie ująć znaczenia elementów (np. ogon, głowa, zderzak)

<https://cs231n.github.io/classification/>



GHOST

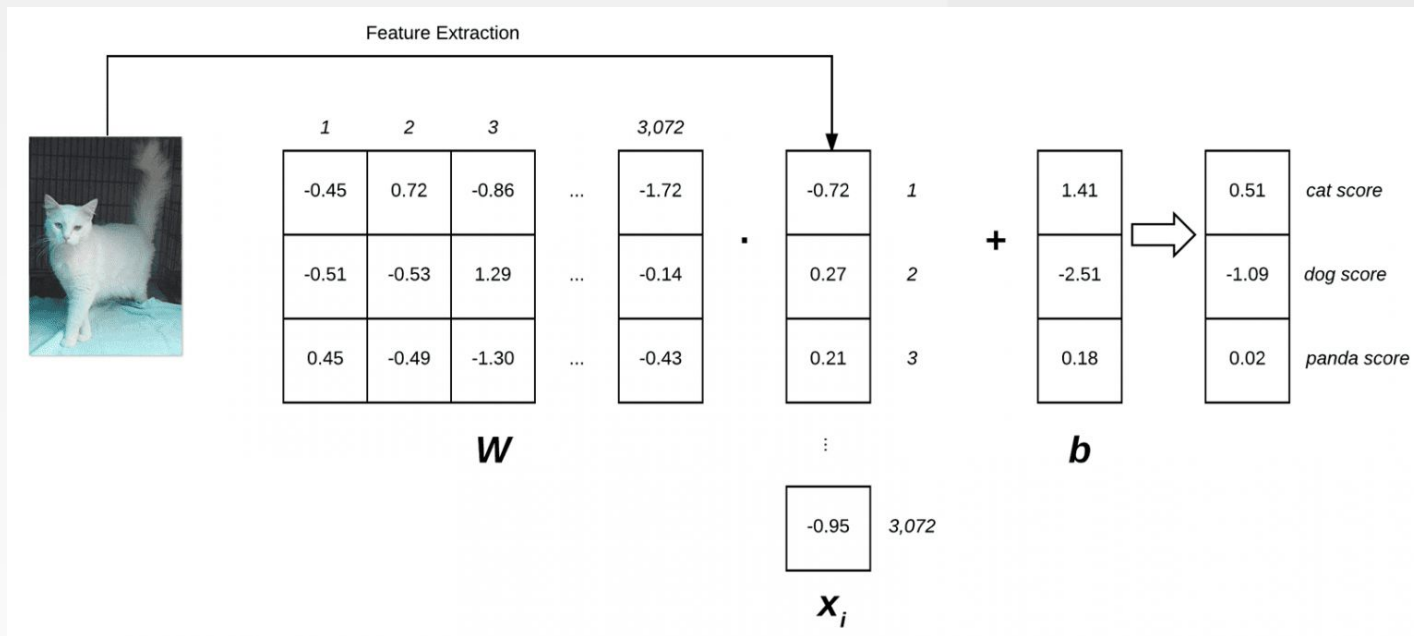
Group of Horribly Optimistic Statisticians

<https://cs231n.github.io/linear-classify/>



Klasyfikator liniowy

$$f(x_i, W) = Wx_i$$

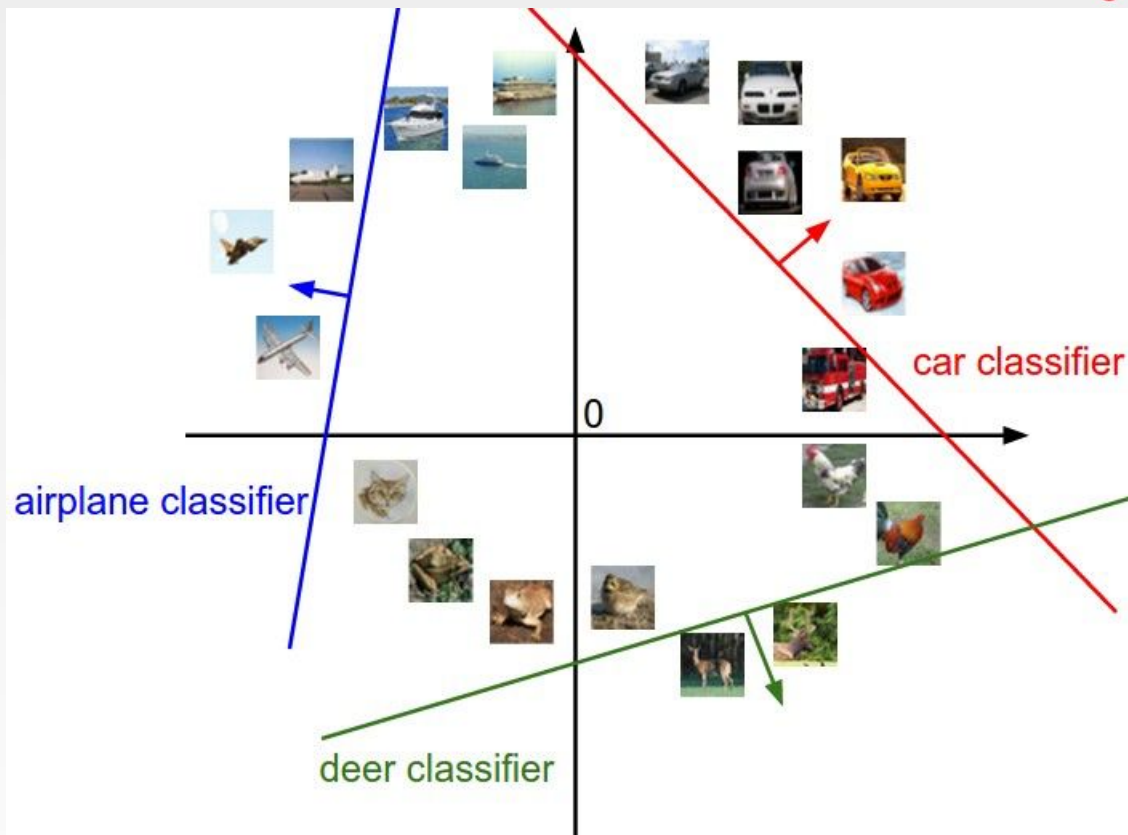




GHOST

Group of Horribly Optimistic Statisticians

Klasyfikator liniowy - intuicja

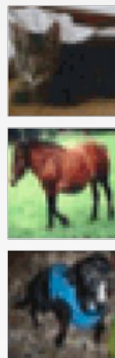


**GHOST**

Group of Horribly Optimistic Statisticians

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \quad \text{or equivalently} \quad L_i = -f_{y_i} + \log \sum_j e^{f_j}$$

Softmax

Input pixels, \mathbf{x} 

Shape: (3, 32, 32)

Forward
propagation →Feedforward output, \mathbf{y}_i

	cat	dog	horse
	5	4	2
	4	2	8
	4	4	1

Shape: (3,)

Softmax
function →Softmax output, $\mathbf{S}(\mathbf{y}_i)$

	cat	dog	horse
	0.71	0.26	0.04
	0.02	0.00	0.98
	0.49	0.49	0.02

Shape: (3,)



GHOST

Group of Horribly Optimistic Statisticians



Parametry jako wzorce (template matching)



Skipping ahead a bit: Example learned weights at the end of learning for CIFAR-10. Note that, for example, the ship template contains a lot of blue pixels as expected. This template will therefore give a high score once it is matched against images of ships on the ocean with an inner product.



GHOST

Group of Horribly Optimistic Statisticians



Klasyfikator liniowy

Zalety

- Prosty wzór
- Szybka inferencja (mnożenie macierzy)
- Nie wymaga przechowywania danych wejściowych w pamięci (przechowujemy jedynie wagi)

Wady

- Jak znaleźć odpowiednie wagi?

<https://cs231n.github.io/linear-classify/>

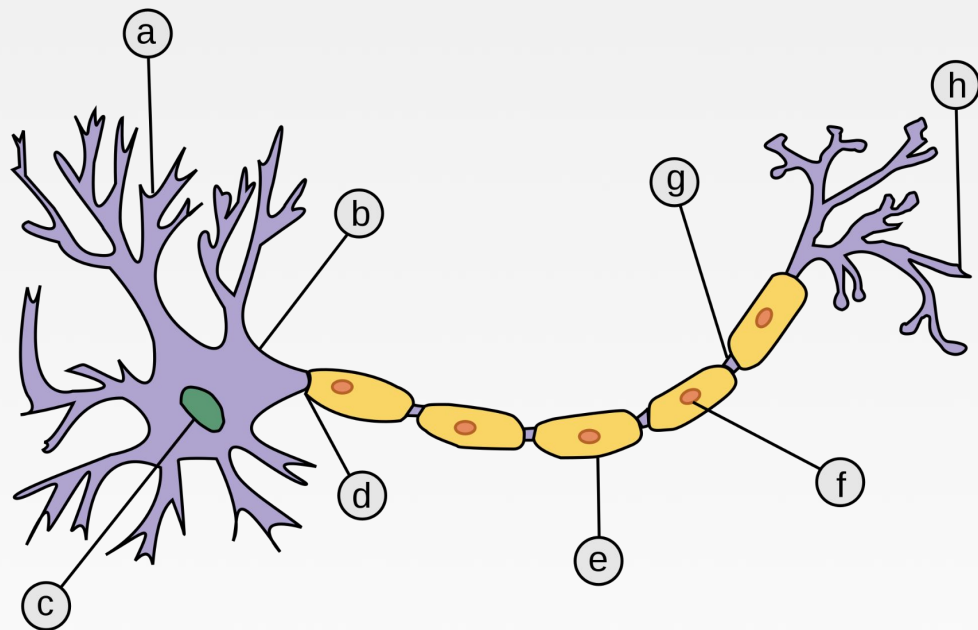


GHOST

Group of Horribly Optimistic Statisticians



Neuron w układzie nerwowym



Schemat budowy neuronu:

a – dendryty,
b – ciało komórki,
c – jądro komórkowe,
d – akson,
e – otoczka mielinowa,
f – komórka Schwanna,
g – przewężenie Ranviera,
h – zakończenia aksonu

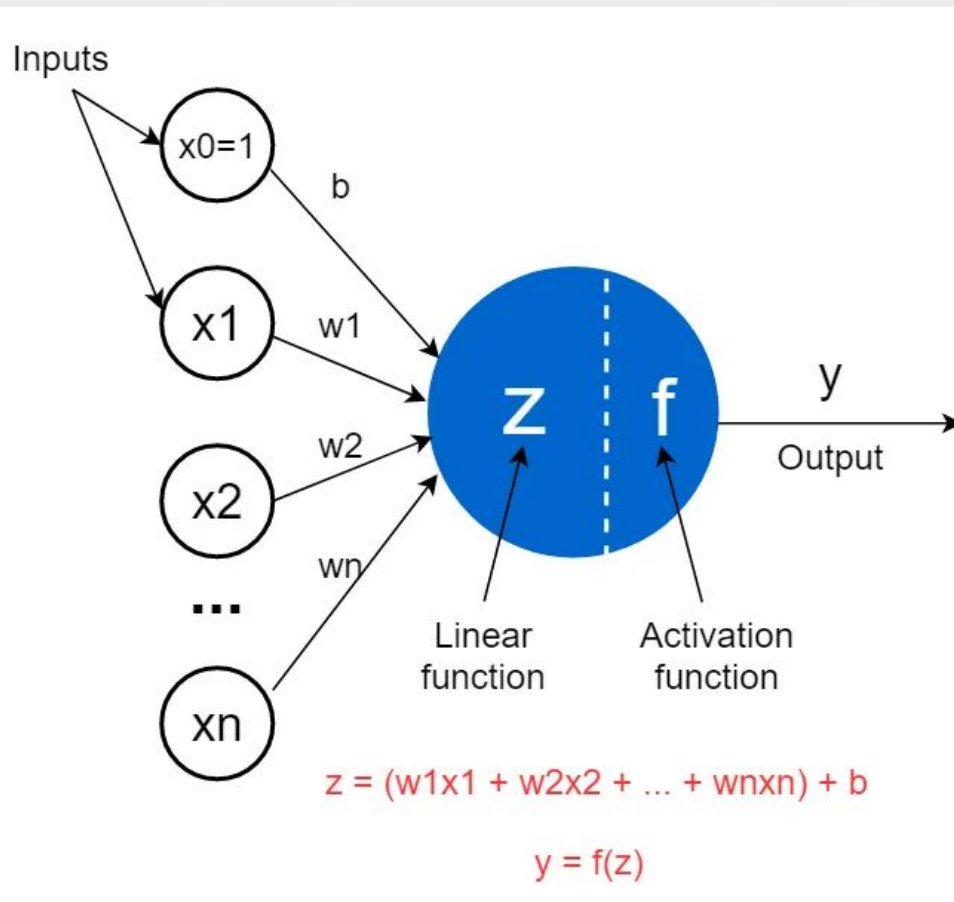


GHOST

Group of Horribly Optimistic Statisticians

Sztuczny neuron

[https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-net-works-fab22249cbfc](https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc)





GHOST

Group of Horribly Optimistic Statisticians

https://en.wikipedia.org/wiki/Activation_function

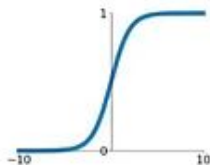


Funkcje aktywacji

Activation Functions

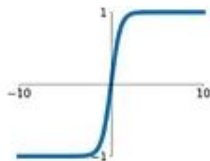
Sigmoid

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



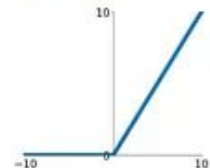
tanh

$$\tanh(x)$$



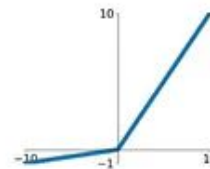
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



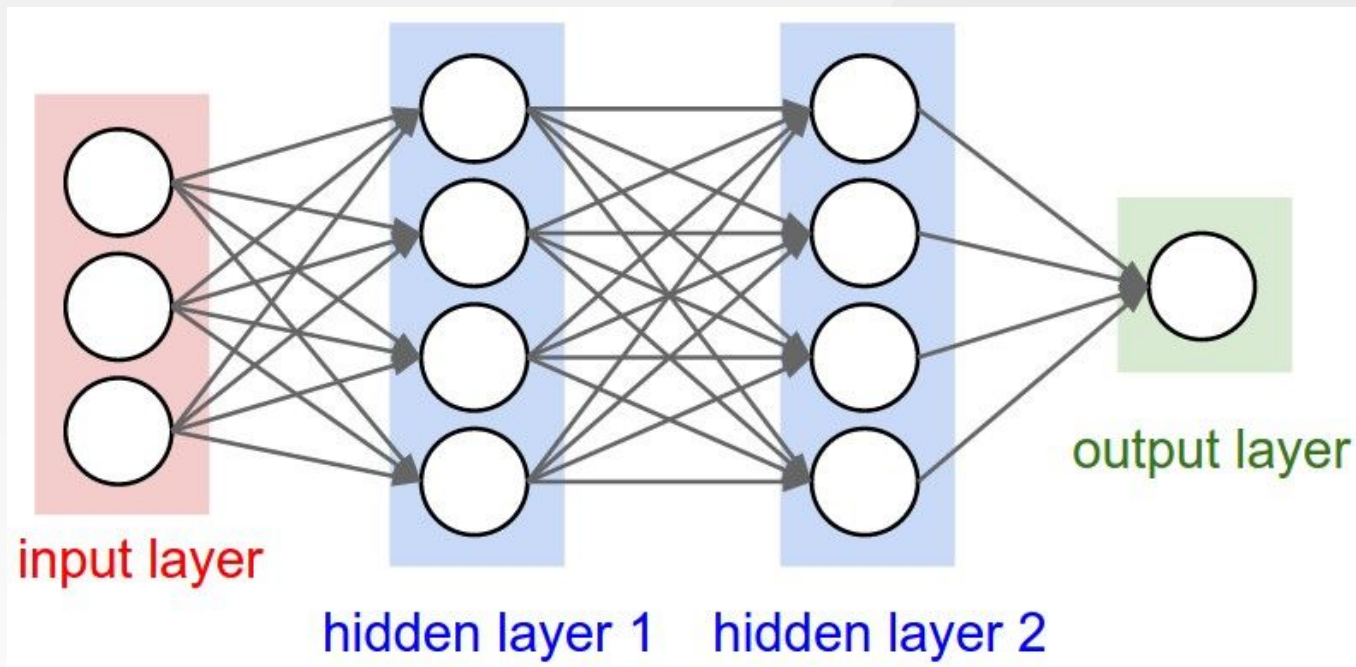


GHOST

Group of Horribly Optimistic Statisticians



Sieć neuronowa jako graf





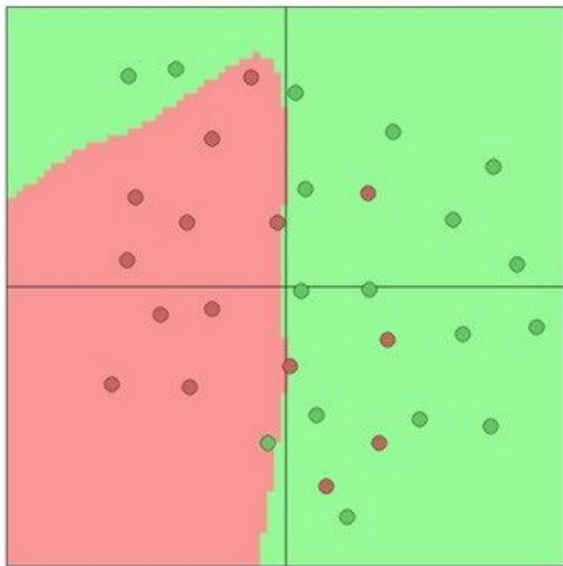
GHOST

Group of Horribly Optimistic Statisticians

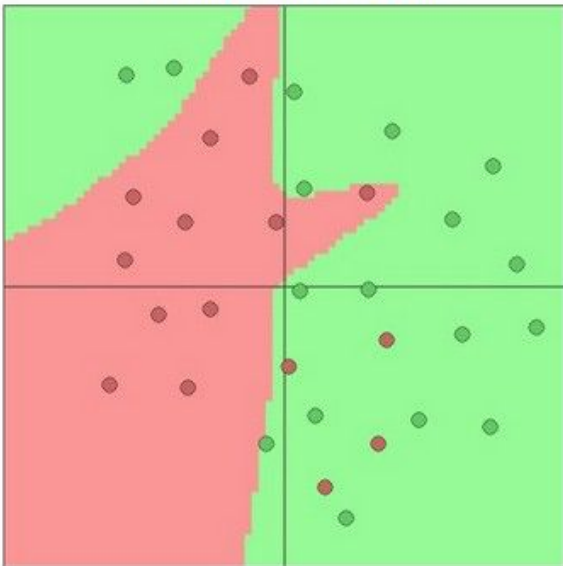


Ilość neuronów a reprezentacja złożonych funkcji

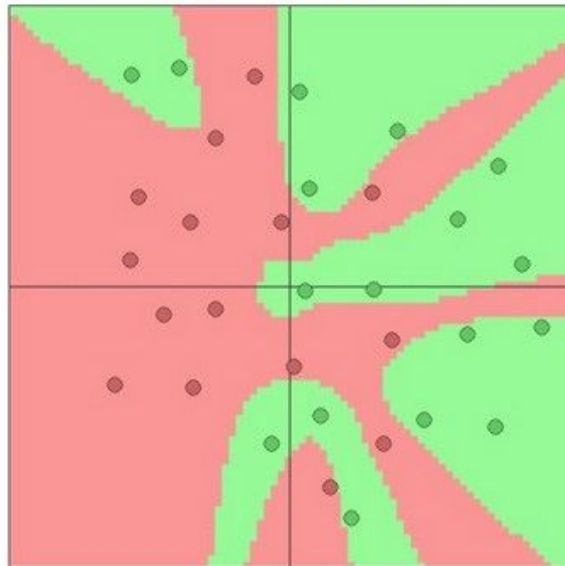
3 hidden neurons



6 hidden neurons



20 hidden neurons



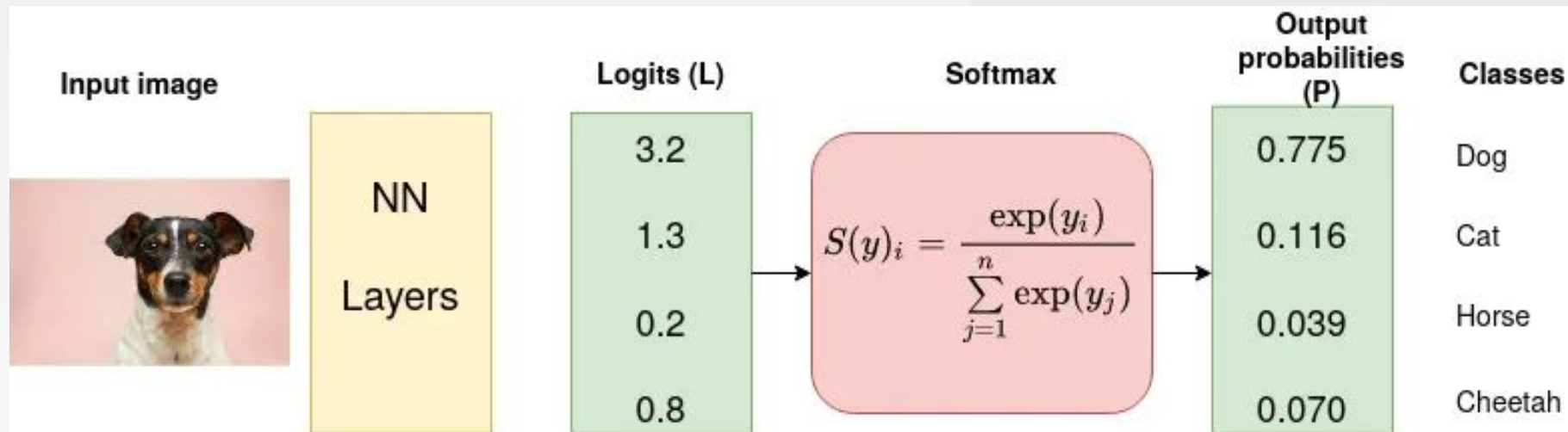


GHOST

Group of Horribly Optimistic Statisticians



Funkcja straty

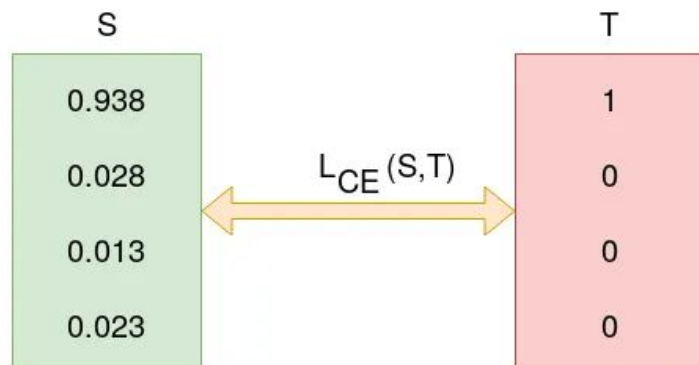


<https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e>

**GHOST**

Group of Horribly Optimistic Statisticians

Entropia krzyżowa



Definicja [[edytuj](#) | [edytuj kod](#)]

Wzór na entropię krzyżową dyskretnej zmiennej losowej X o zbiorze wartości $\{x_1, x_2, \dots, x_n\}$ i zdefiniowanych na nich rozkładach prawdopodobieństw p i q ^[1]:

$$H(p, q, X) = - \sum_{i=1}^n p(x_i) \log q(x_i).$$

Interpretując entropię krzyżową jako miarę nietrafności, za p przyjmuje się rozkład traktowany jako prawdziwy, a za q rozkład oceniany^[2].

https://pl.wikipedia.org/wiki/Entropia_krzy%C5%BCowa



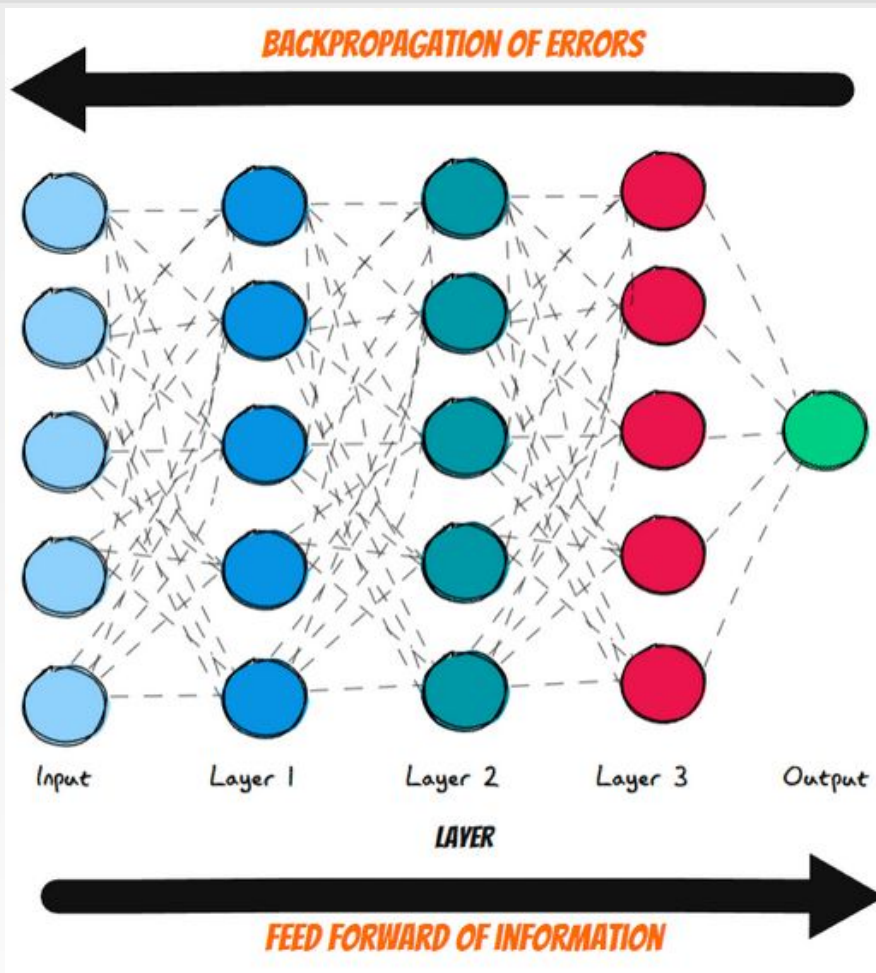
GHOST

Group of Horribly Optimistic Statisticians

Propagacja wsteczna

$$*W_x = W_x - \underset{\substack{\uparrow \\ \text{Learning} \\ \text{rate}}}{a} \left(\underset{\substack{\uparrow \\ \text{Derivative of Error} \\ \text{with respect to weight}}}{\frac{\partial \text{Error}}{\partial W_x}} \right)$$

\uparrow New weight





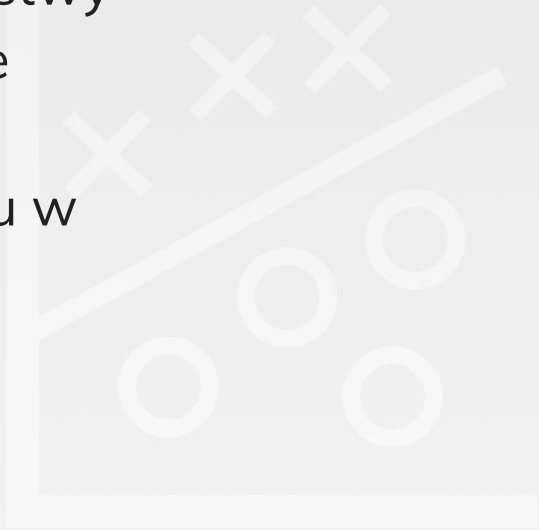
GHOST

Group of Horribly Optimistic Statisticians



Sieć neuronowa - podsumowanie

- Złożone z wielu neuronów ułożonych w warstwy
- Wagi (parametry) aktualizowane w procesie iteracyjnym
- Funkcja straty określa jakość wyjścia modelu w stosunku do wartości oczekiwanej





GHOST

Group of Horribly Optimistic Statisticians



Możliwość przybliżenia wartości funkcji

It turns out that Neural Networks with at least one hidden layer are *universal approximators*. That is, it can be shown (e.g. see [Approximation by Superpositions of Sigmoidal Function](#) from 1989 (pdf), or this [intuitive explanation](#) from Michael Nielsen) that given any continuous function $f(x)$ and some $\epsilon > 0$, there exists a Neural Network $g(x)$ with one hidden layer (with a reasonable choice of non-linearity, e.g. sigmoid) such that $\forall x, |f(x) - g(x)| < \epsilon$. In other words, the neural network can approximate any continuous function.

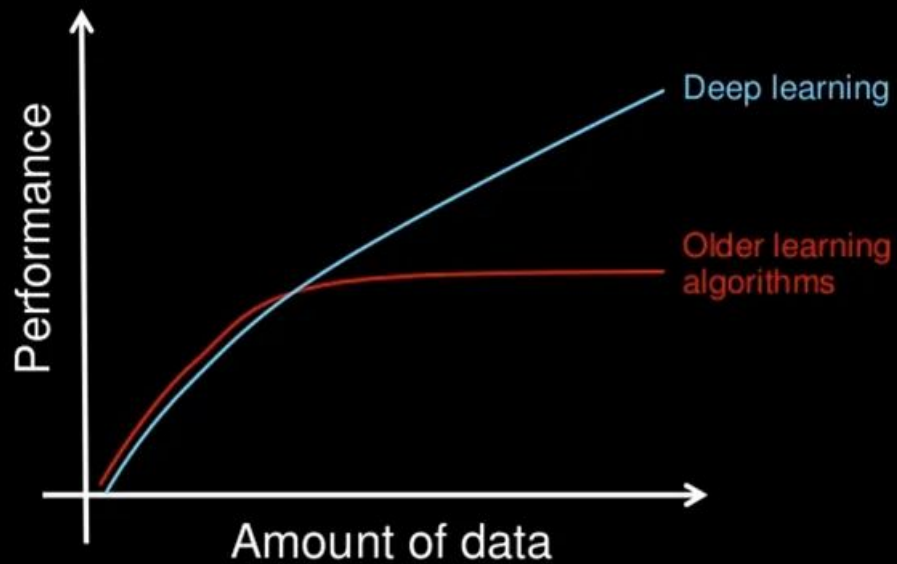


GHOST

Group of Horribly Optimistic Statisticians



Why deep learning





GHOST

Group of Horribly Optimistic Statisticians



Materialy

- A visual proof that neural nets can compute any function

<http://neuralnetworksanddeeplearning.com/chap4.html>

- ConvNetJS

<https://cs.stanford.edu/people/karpathy/convnetjs/>

- A Step by Step Backpropagation Example

<https://mattmazor.com/2015/03/17/a-step-by-step-backpropagation-example/>

- BackPropagation and Gradient Descent

- <https://colab.research.google.com/github/jigsawlabs-student/pytorch-intro-curriculum/blob/main/5-training-mathematically/20-backpropagation-and-gradient-descent.ipynb>

- Backprop example in colab

- <https://colab.research.google.com/github/Alexander-Schiendorfer/Alexander-Schiendorfer.github.io/blob/master/notebooks/backprop-example.ipynb>

