

# WARSZTATY BADAWCZE DEEP LEARNING-2

**Paulina Tomaszewska**

**[p.tomaszewska2@student.mini.pw.edu.pl](mailto:p.tomaszewska2@student.mini.pw.edu.pl)**

**albo czat na Ms Teams**

# ZASADY

- Nie ma głupich pytań
- Staramy się mieć włączone kamerki
- Regularne informowanie o statusie prac

# GŁÓWNE ZAŁOŻENIA PRZEDMIOTU

- Wysokopoziomowe zrozumienie sieci neuronowych  
(inne akcenty niż na klasycznym kursie z DL)
- Umiejętność poszukiwania informacji i wnioskowania + pisanie artykułu
- Dobre nawyki w pracy badawczej (reprodukowalność)

# FORMUŁA

- *inverse learning* bo praca badawcza to m.in. poszukiwanie informacji (dyskusje)
- Książka: Deep Learning with Python (by F. Chollet):  
<http://faculty.neu.edu.cn/yury/AI/Textbook/Deep%20Learning%20with%20Python.pdf>
- Jupyter notebooks: <https://github.com/fchollet/deep-learning-with-python-notebooks>

# PLAN PRAC

## Prace domowe:

- P1 – zadanie-rozgrzewka (2 pkt.)
- P2 – warstwa konwolucyjna (6pkt.)
- P3 – zmienna wielkość danych wejściowych (6pkt.)
- P4 – *ablation study* + dodać inną miarę jakości\* (4pkt.)
- P5 – *sensitive analysis* + krótko o niepewnościach w sieciach\* (4pkt.)
- P6 – *lime* + porównanie z *GRAM\_CAD*\* (6pkt.)

## Kamienie milowe (motywacja do systematycznej pracy):

- KM1 – *motivation&network summary* (5pkt.)
- KM2 – *results of reproducibility*: co zostało zrobione i z jakim efektem? (6pkt.)
- +1 pkt. za napisanie recenzji

\*na bazie kodu z projektu lub innego

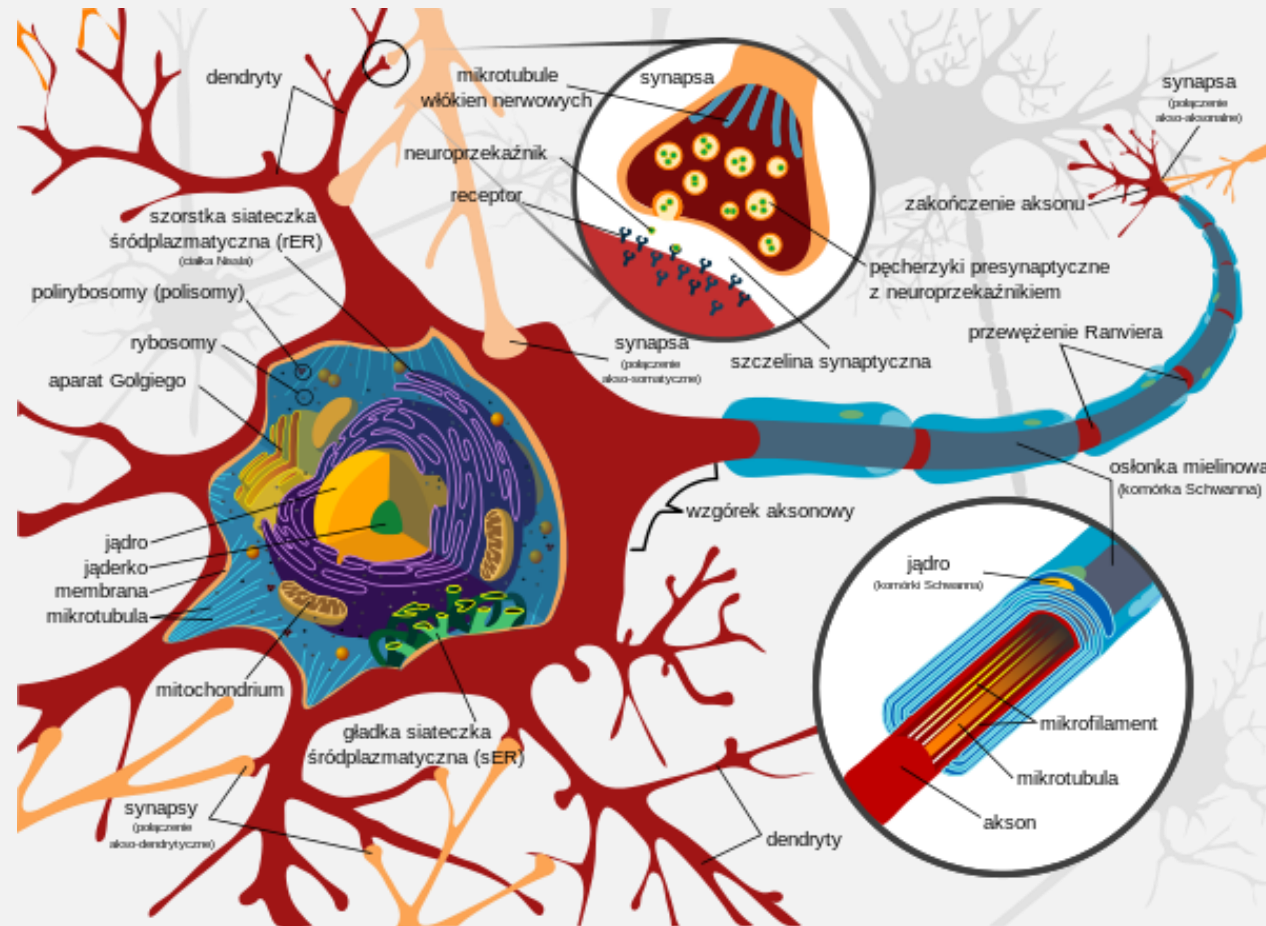
# GOOGLE COLABORATORY

- Live demo

# ZADANIE NA ROZGRZEWKĘ – „KREATYWNOŚĆ BADACZA”

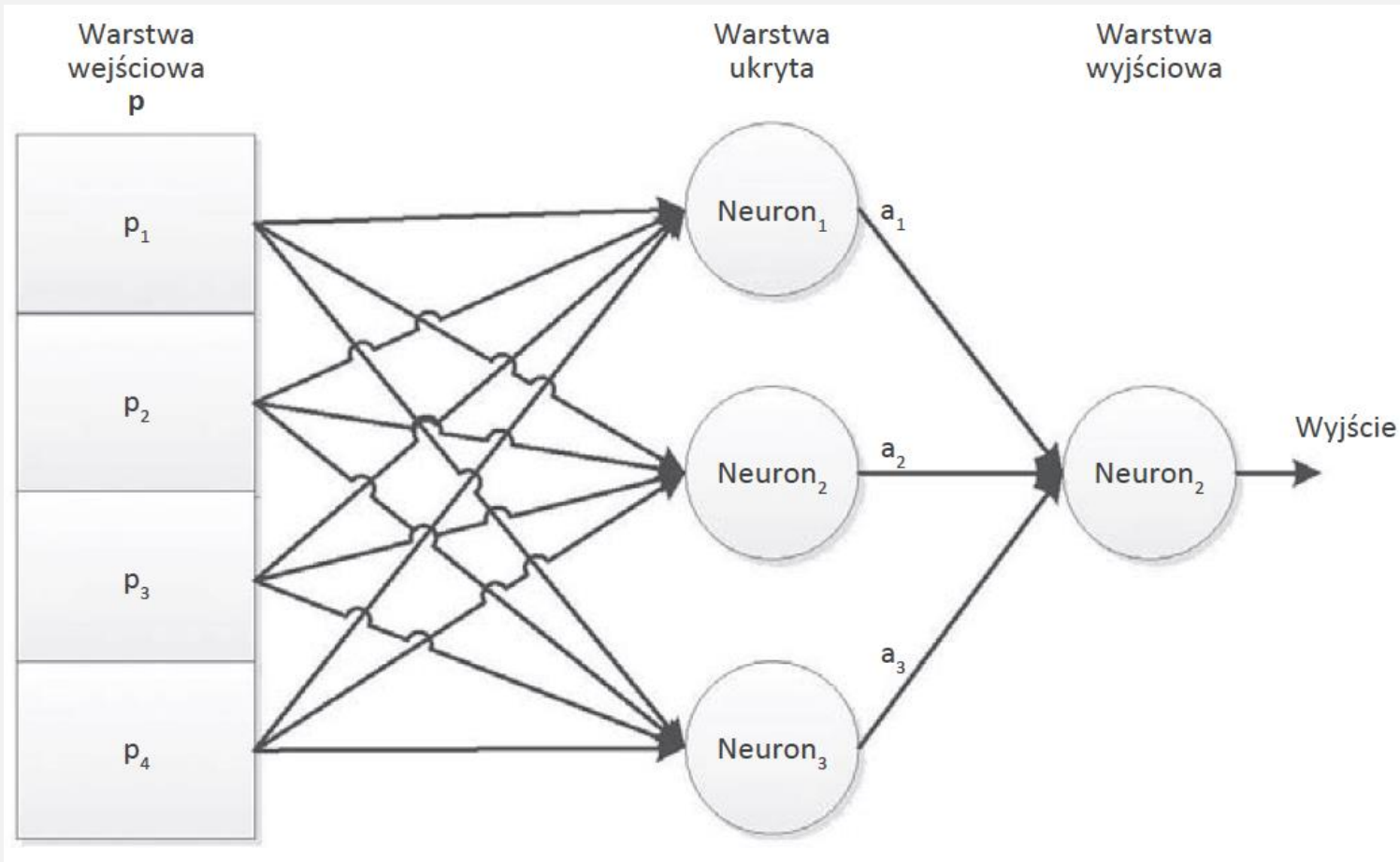
- Zbiór danych: <https://archive.ics.uci.edu/ml/datasets/Student+Performance>
- Postawić 5 hipotez i zdecydować czy należy je odrzucić czy nie ma podstaw (nie są wymagane testy statystyczne)
- Jupyter Notebook + wygenerowany html

# NEURONY



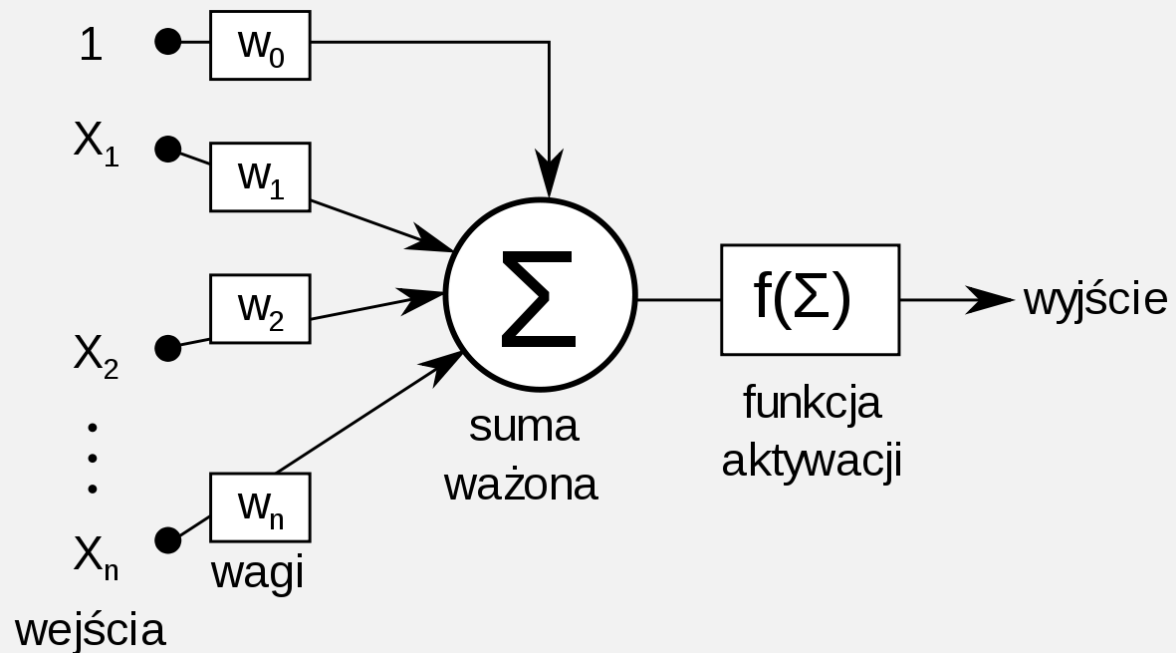


# SIEĆ NEURONOWA – MULTILAYER PERCEPTRON



*Ile powinno być warstw ukrytych/ neuronów?*

# SZTUCZNY NEURON

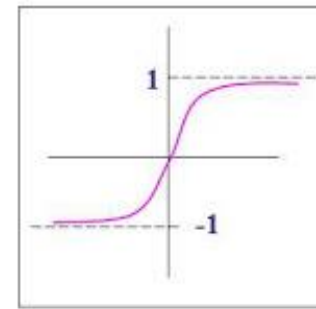


[https://pl.wikipedia.org/wiki/Neuron\\_McCullocha-Pittsa](https://pl.wikipedia.org/wiki/Neuron_McCullocha-Pittsa)

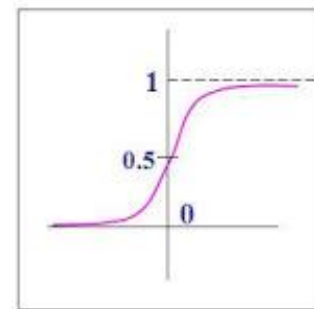
*suma ważona*

$$= w_0 * 1 + w_1 * x_1 + w_2 * x_2 + \dots$$

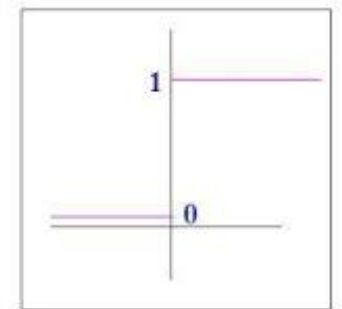
Funkcja aktywacji



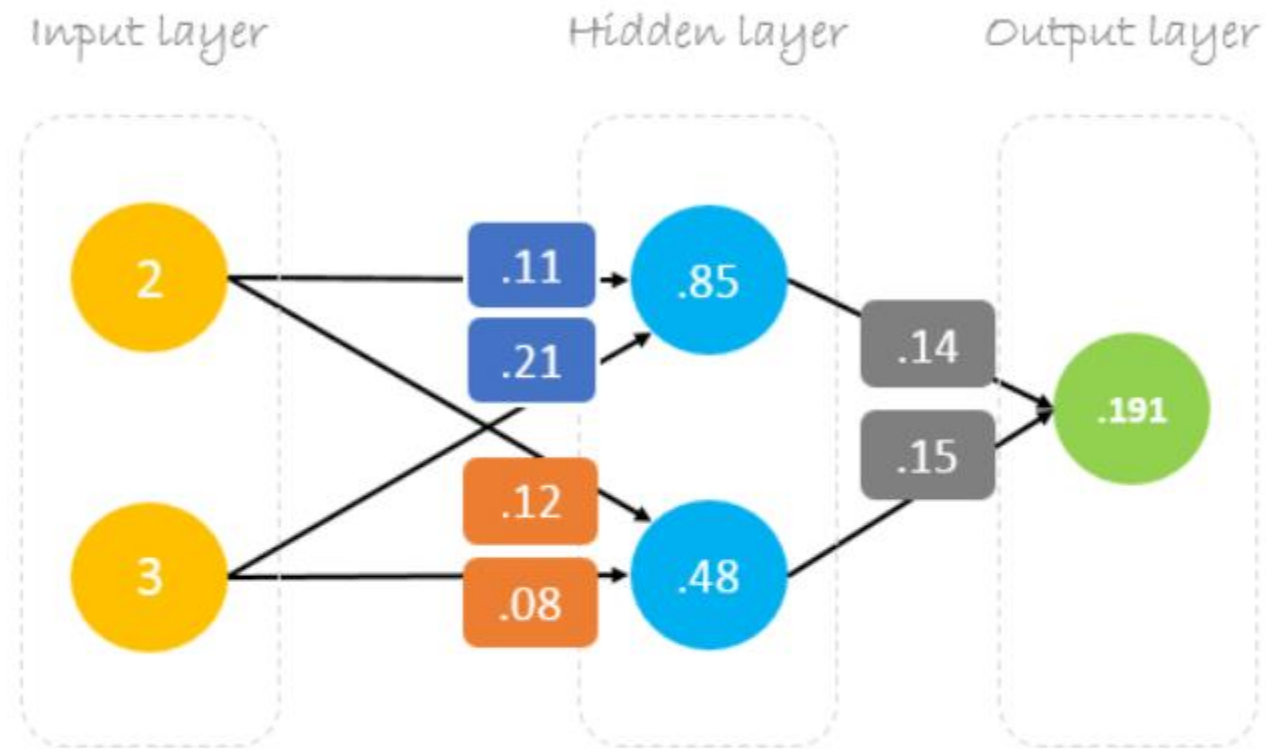
**Tanh**  
 $f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$



**Logistyczna**  
 $f(x) = 1 / (1 + e^{-x})$



**Skok jedn.**  
 $f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$



Forward Pass

$$\begin{bmatrix} 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0.11 & 0.12 \\ 0.21 & 0.08 \end{bmatrix} = \begin{bmatrix} 0.85 & 0.48 \end{bmatrix} \cdot \begin{bmatrix} 0.14 \\ 0.15 \end{bmatrix} = \begin{bmatrix} 0.191 \end{bmatrix}$$

Matrix multiplication

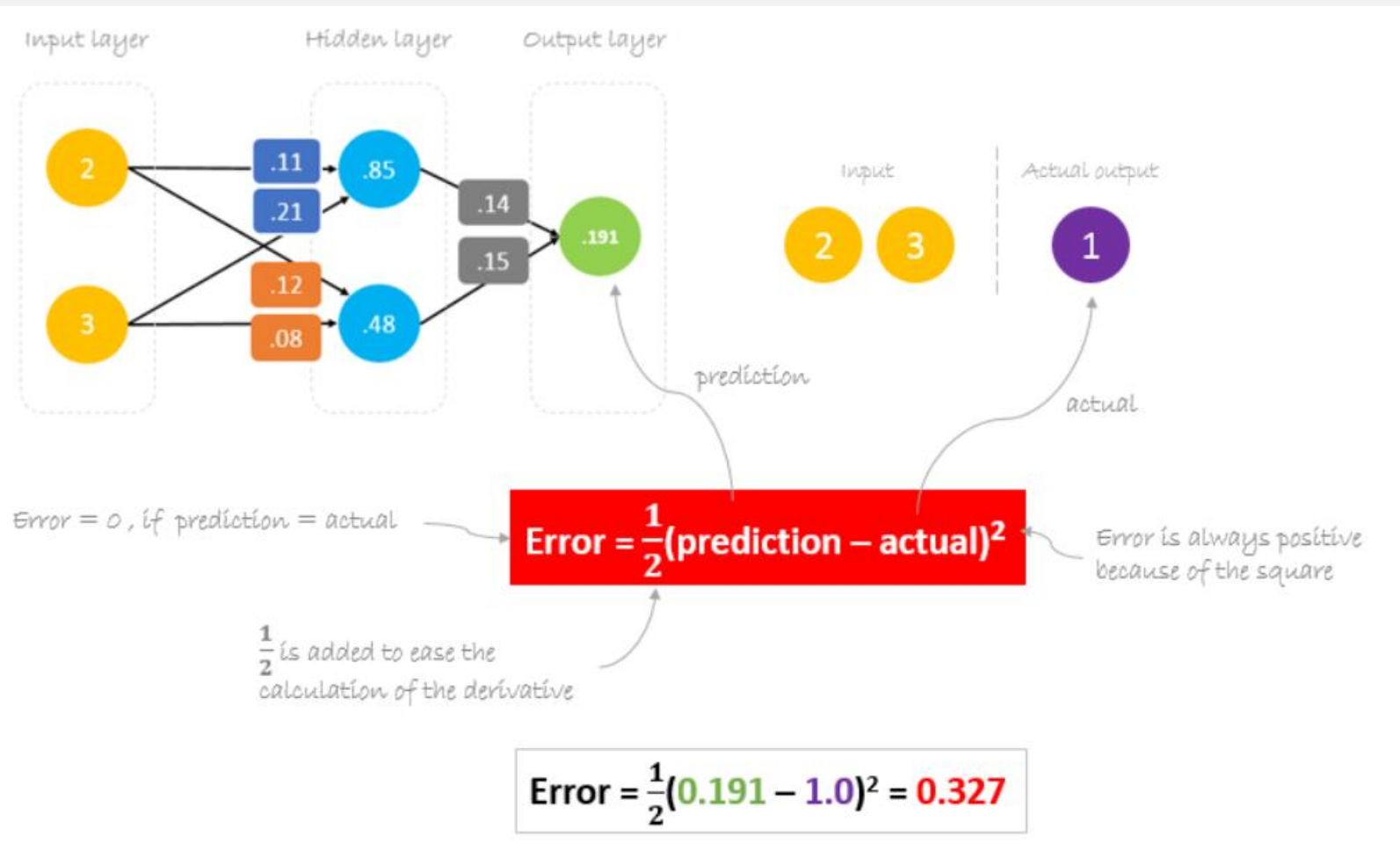
Details

$$2 \times .11 + 3 \times .21 = .85$$

$$.85 \times .14 + .48 \times .15 = .191$$

$$2 \times .12 + 3 \times .08 = .48$$

# ALGORYTM PROPAGACJI WSTECZNEJ



Ważny parametr: **learning rate**  
(odpowiada za szybkość treningu)

$$w_{t+1} = w_t - \text{learning\_rate} * \text{błąd}$$

Jak długo trwa trening?

$$\frac{\partial E}{\partial W}$$

## CZAS NA LEPSZE ZROZUMIENIE

- <https://playground.tensorflow.org/>