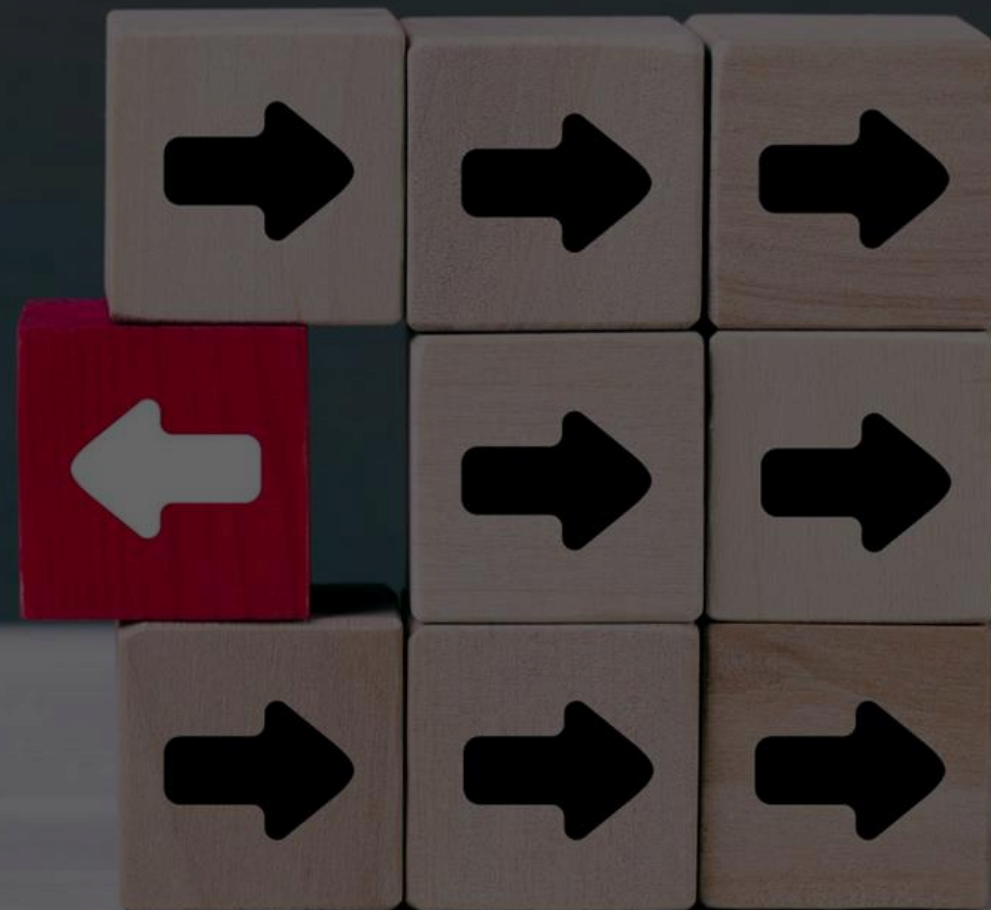


WARSZTATY BADAWCZE: „METODY PRZENOSZENIA WIEDZY W SIECIACH NEURONOWYCH”

-- SIECI MLP --



Paulina Tomaszewska

Poznajmy się (aby potem lepiej działać w zespołach)

1. Imię
2. Przymiotnik zaczynający się od pierwszej litery imienia
3. Doświadczenie z sieciami neuronowymi

Zasady

- Nie wahamy się z zadawaniem merytorycznych pytań
- Regularne informowanie o statusie prac projektowych
- Formuła zajęć uwzględniająca prezentacje studentów, dyskusje, recenzje (co stanowi przygotowanie do finalnej prezentacji projektu)
- Projekty i prace domowe można realizować w Keras lub Pytorch
- Na zajęciach omawiane będą zastosowania z dziedziny analizy obrazów
- Repozytorium: <https://github.com/MI2-Education/2022L-WB-TL>
- Materiały w większości w języku angielskim

Główne założenia przedmiotu

- Wysokopoziomowe zrozumienie sieci neuronowych
- Główny nacisk na metody przenoszenia wiedzy (umiejętność implementacji rozwiązań z artykułu)
- Umiejętność poszukiwania obszaru badawczego, zbierania informacji i wnioskowania (*inverse learning*), pisania artykułu (imitacja całego procesu badawczego)
- Dobre nawyki w pracy badawczej
- Początkowo omawiany jest framework Keras a potem przez analogię Pytorch

Projekt

- Implementacja rozwiązania z artykułu nt. przenoszenia wiedzy w sieciach neuronowych – dane obrazowe:
 - *Opcja 1: artykuł dotyczy danych ogólnego przeznaczenia, dokonywana jest implementacja i porównanie się z wynikami z artykułu, potem następuje adaptacja do danych medycznych*
 - *Opcja 2: artykuł dotyczy danych medycznych, dokonywana jest implementacja najpierw dla danych ogólnego przeznaczenia a następnie dla danych medycznych (wówczas porównanie się z wynikami z artykułu)*

Implementowane rozwiązanie nie może być z kanonu rozwiązań z dziedziny, przenoszenia wiedzy w sieciach neuronowych (wyjątkiem jest gdy popularne rozwiązanie jest wzbogacone o nowe elementy).

Literatura

- Wprowadzenie do sieci neuronowych:
 - *Książka: Deep Learning with Python (by F. Chollet):*
<https://www.manning.com/books/deep-learning-with-python>
 - *Jupyter notebooks:* <https://github.com/fchollet/deep-learning-with-python-notebooks>
- Tematyka przenoszenia wiedzy w sieciach neuronowych
 - *Książka: Hands-On Meta Learning with Python:*
<https://www.packtpub.com/product/hands-on-meta-learning-with-python/9781789534207> (pierwszy rozdział dostępny online)
 - *Jupyter notebooks:* <https://github.com/PacktPublishing/Hands-On-Meta-Learning-with-Python>
 - <https://sites.google.com/mit.edu/aaai2021metalearningtutorial/home>

Punktacja

- praca nad projektem (48 pkt.):
 - kamień milowy_1a: wstępny przegląd literatury z wybranego obszaru transferu wiedzy, harmonogram prac i podział obowiązków, prezentacja (11 pkt.)*
 - aktywność_1: przygotowanie i poprowadzenie debaty studenckiej (3 pkt.)
 - kamień milowy_1b: wstępny opis literatury w ramach artykułu (5 pkt.)*
 - kamień milowy_2: (17 pkt. = 10 pkt. + 7 pkt.*)
 - aktywność_2: opracowanie na zajęciach jednego z najnowszych artykułów na Arxiv pod kątem (Why?, What?, How?), prezentacja (5 pkt.)
 - aktywność_3: recenzja prezentacji w ramach aktywności 2 (2 pkt.)
 - aktywność_4: recenzja prezentacji finalnych (5 pkt.)
- prezentacja końcowa (16 pkt.)*
- raport końcowy (32 pkt.)*
- stosowanie dobrych praktyk wykorzystania GitHub (4 pkt.)*

* rozliczane grupowo

Harmonogram

Temat
Kwestie organizacyjne (plan zajęć, system oceniania). Wstęp do sieci neuronowych - architektura MLP.
Pojęcie transferu wiedzy w sieciach neuronowych (szersze niż "transfer learning"). Kod ilustrujący wykorzystanie architektury MLP oraz zjawisko overfittingu.
Idea i kod sieci konwolucyjnych, w tym popularne architektury: Resnet, VGG, DenseNet oraz zagadnienia augmentacji danych.
Framework Keras w wersji niskopoziomowej + analogia do Pytorcha (w tym pytorch.lightning). Wady i zalety dwóch frameworków na przykładzie Relation Network.
Debaty studenckie na tematy z obszaru transferu wiedzy w sieciach neuronowych.
Prezentacje w ramach pierwszego kamienia milowego.
Algorytmy: MLP-Mixer, Knowledge Distillation - skeptical student.
Algorytm Multiple-instance learning.
Algorytm adaptacji w przypadku continual learning.
Algorytmy: Cross-stitch networks, GradNorm.
Sieci typu Autoenkoder, GAN, U-net, wyjaśnienia sieci neuronowych, active learning (wysokopoziomowe wyjaśnienie).

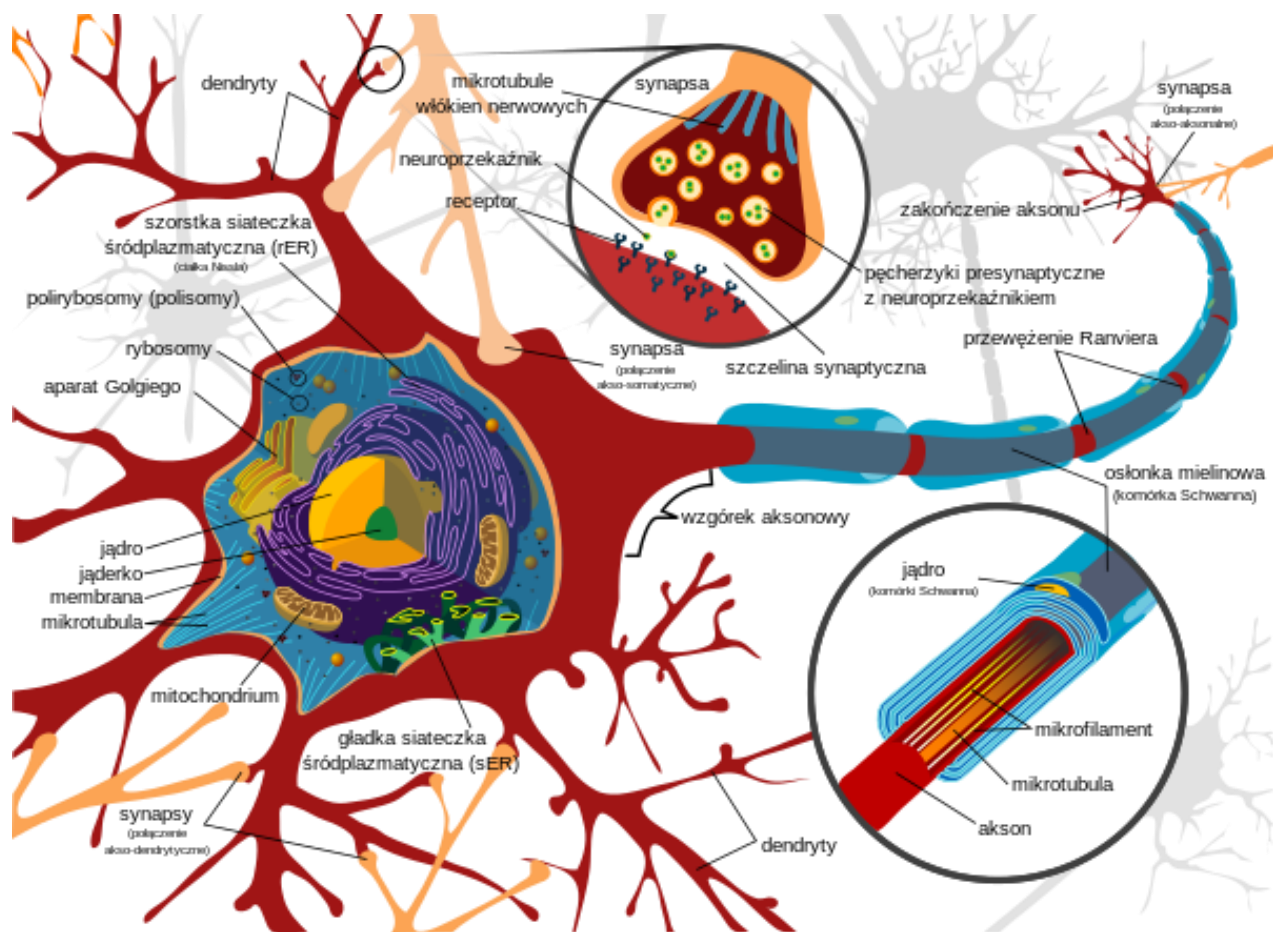
Prezentacje studentów nt. najnowszych artykułów z Arxiv z obszaru transferu wiedzy.

Konsultacje projektowe.

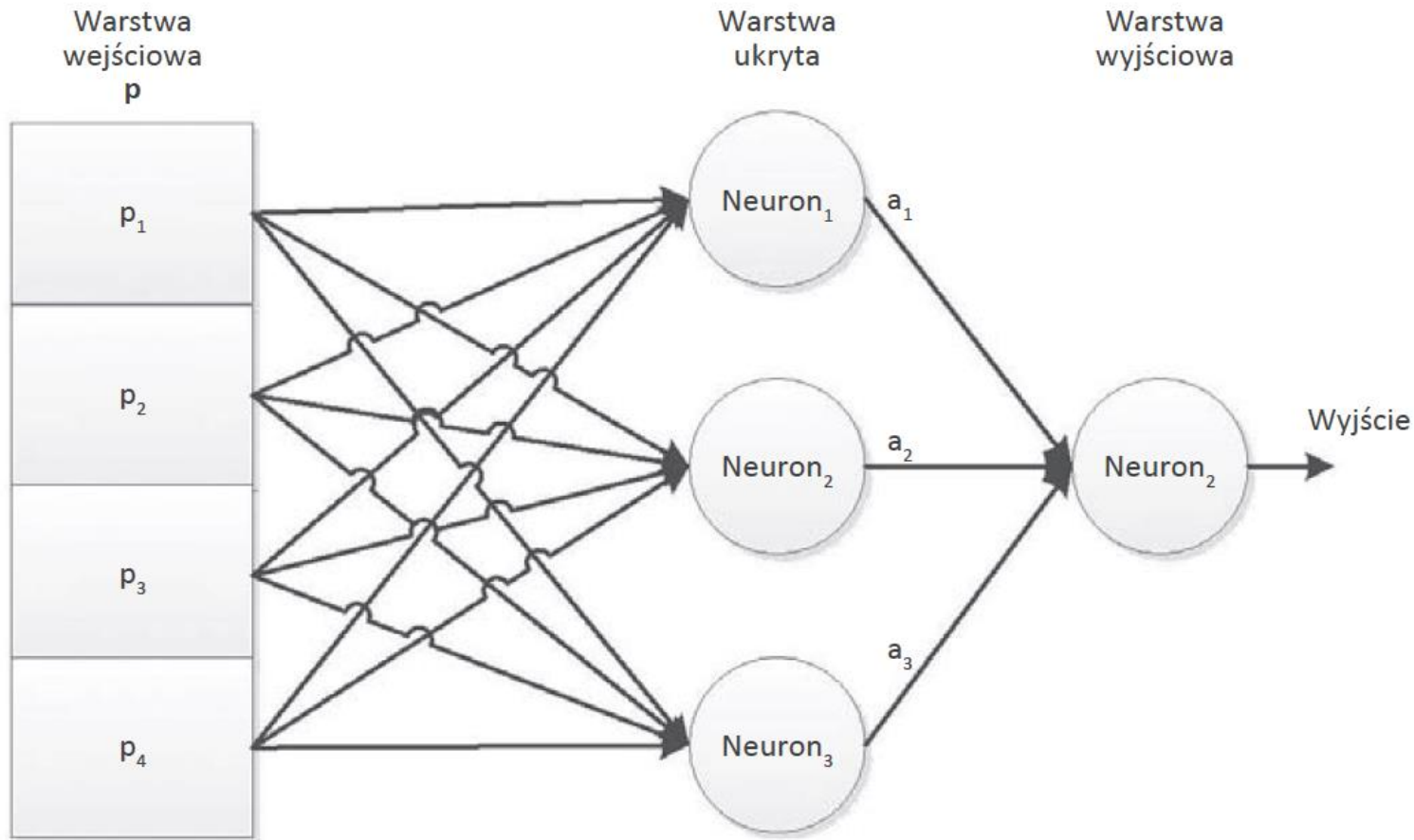
Konsultacje projektowe.

Podsumowanie projektu/konsultacje.

Neurony



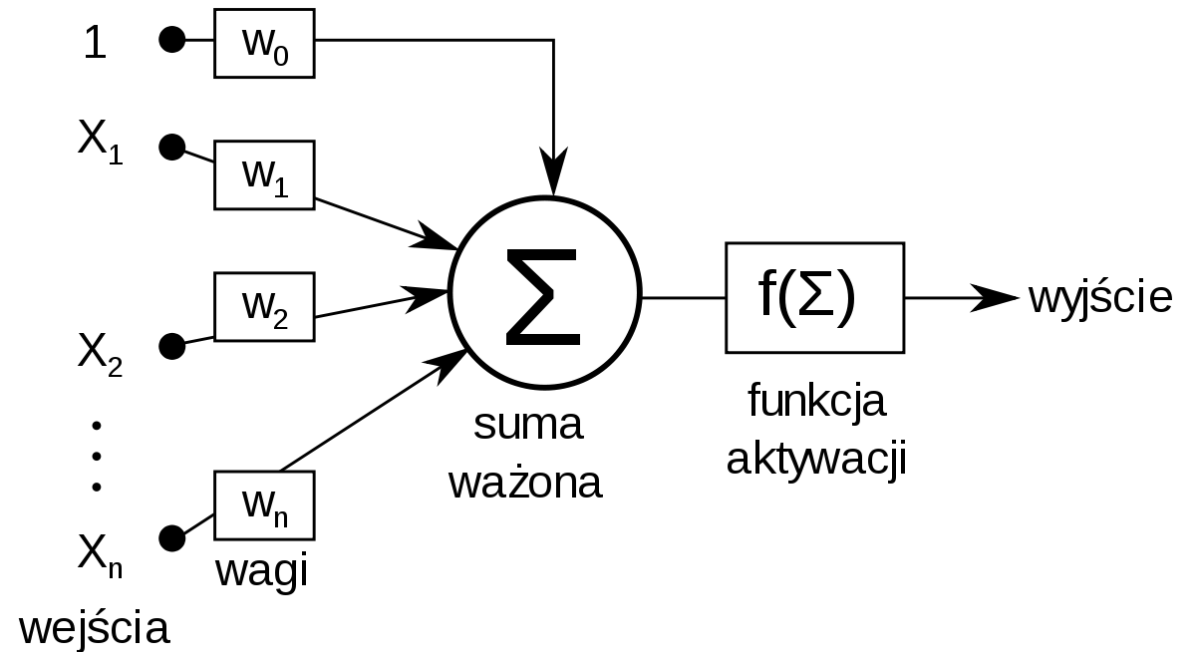
Sieć neuronowa – Multilayer Perceptron (MLP)



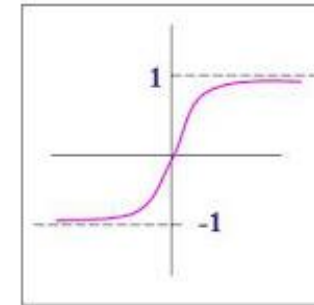
Ile powinno być warstw ukrytych/ neuronów?

Sztuczny neuron

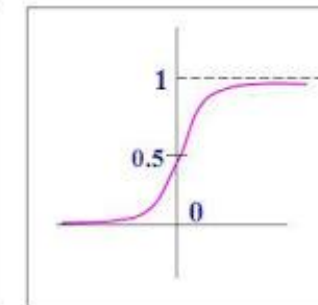
$$\text{suma ważona} = w_0 * 1 + w_1 * x_1 + w_2 * x_2 + \dots$$



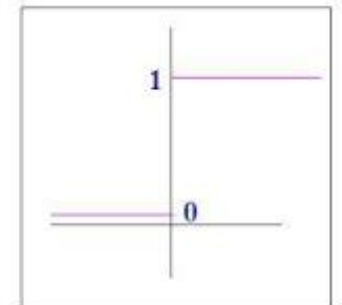
Funkcja aktywacji



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

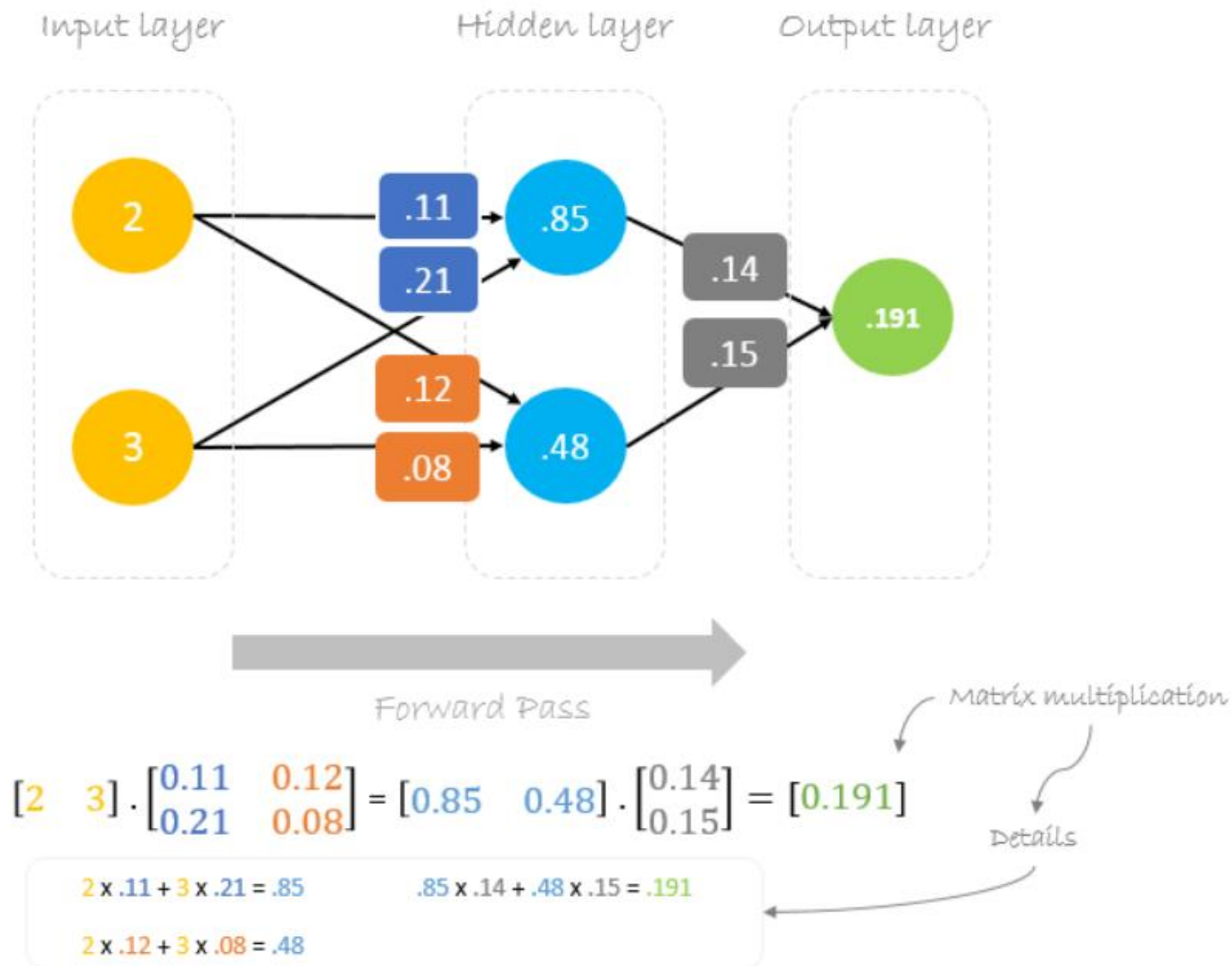


Logistyczna
 $f(x) = 1 / (1 + e^{-x})$

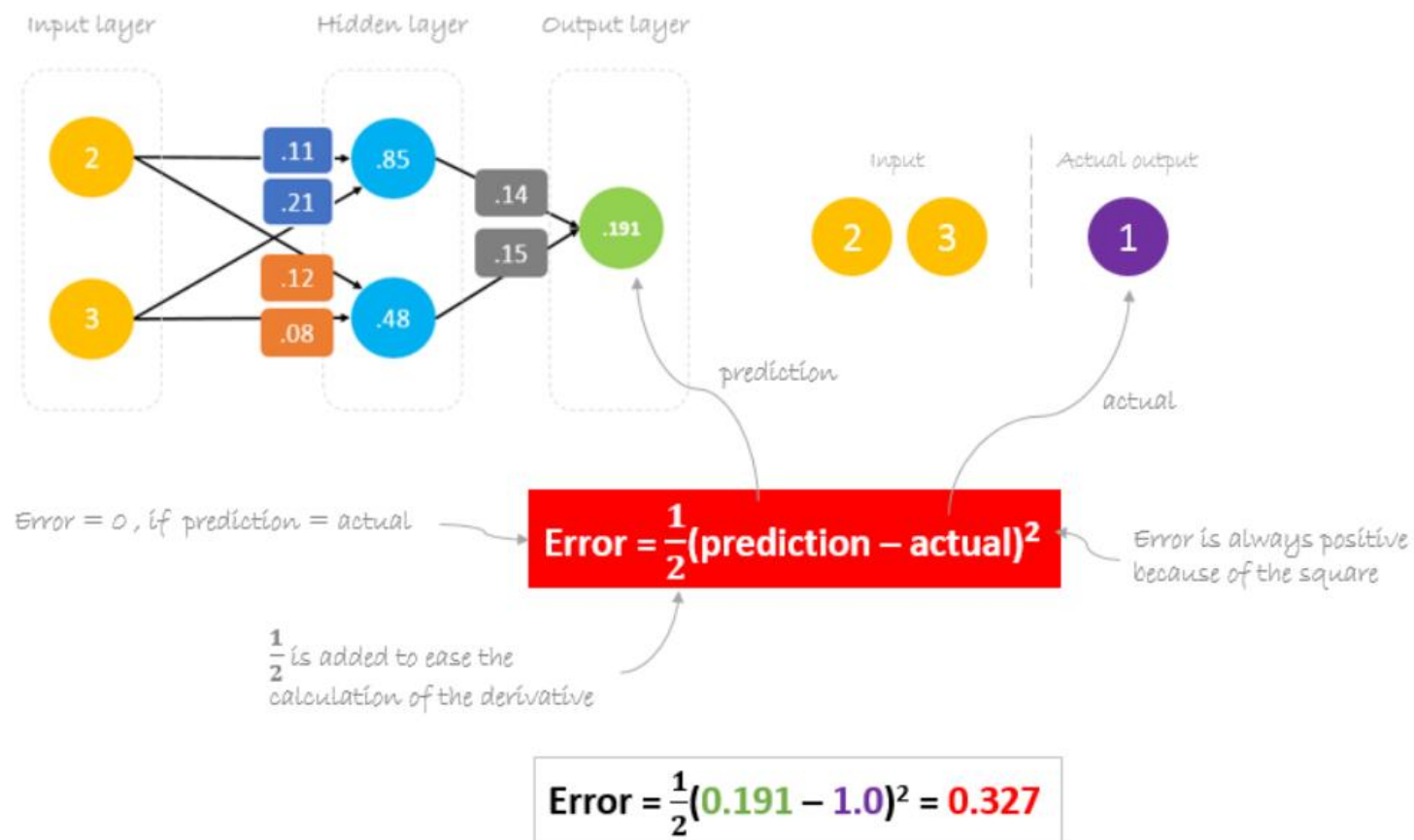


Skok jedn.
 $f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$

Inferencia (forward pass)



Algorytm propagacji wstecznej (backpropagation)



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

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

Jak długo trwa trening?

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

chain rule

Czas na lepsze wyczucie

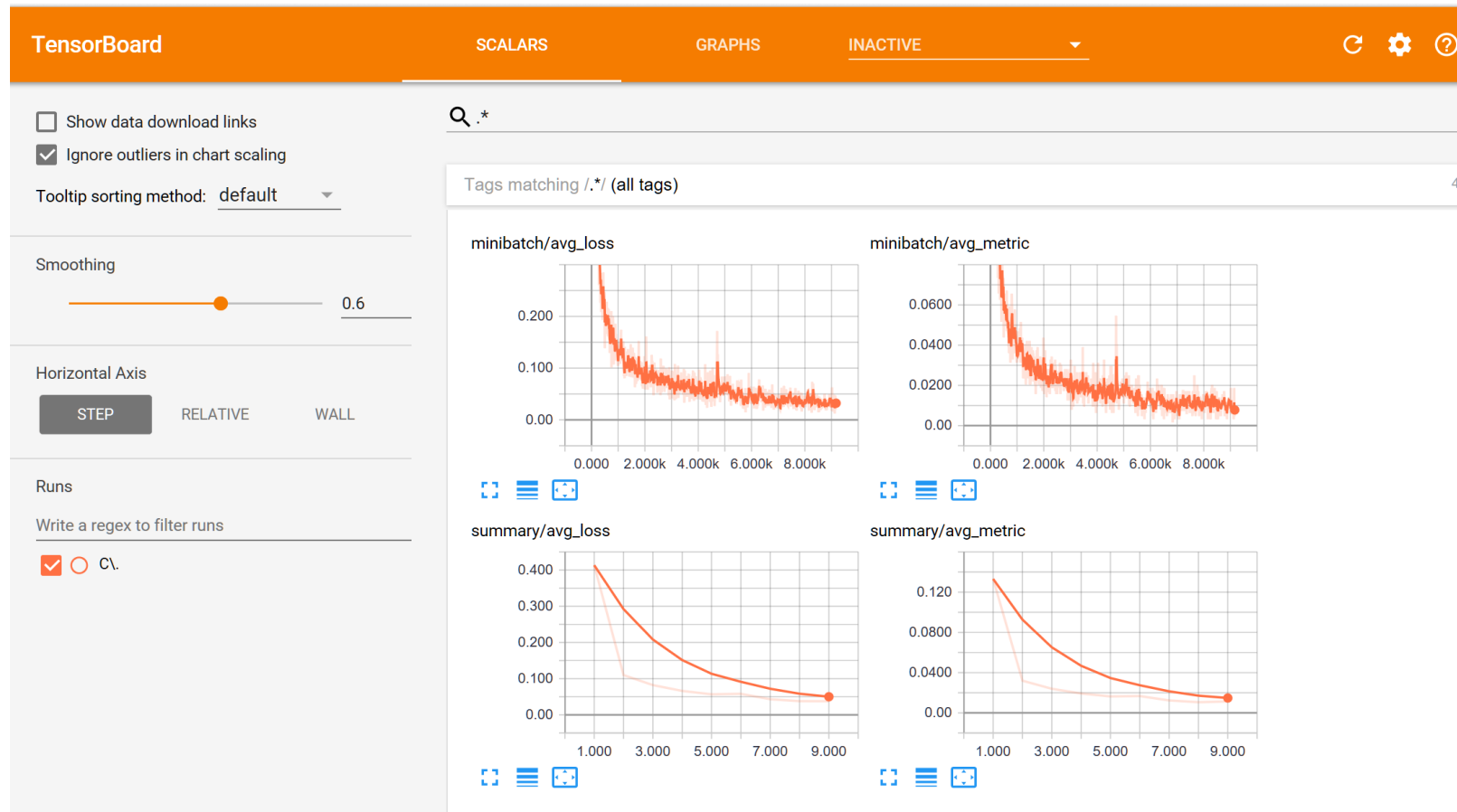
- <https://playground.tensorflow.org/>

Omówienie kodu

- https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/first_edition/2.1-a-first-look-at-a-neural-network.ipynb

TensorBoard

- Używany niezależnie od framework'u



- Tutorial: <https://www.datacamp.com/community/tutorials/tensorboard-tutorial>