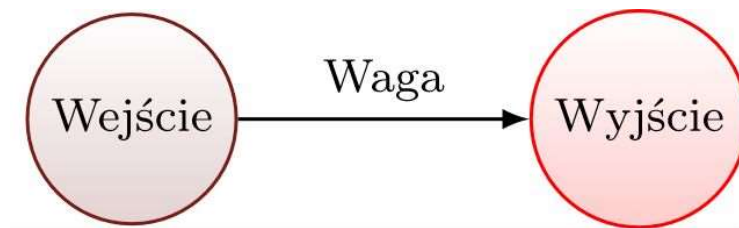


Laboratorium 1

Celem laboratorium jest zapoznanie się z zasadą działania neuronu oraz budowa prostej sieci neuronowej.

Zadanie 1:

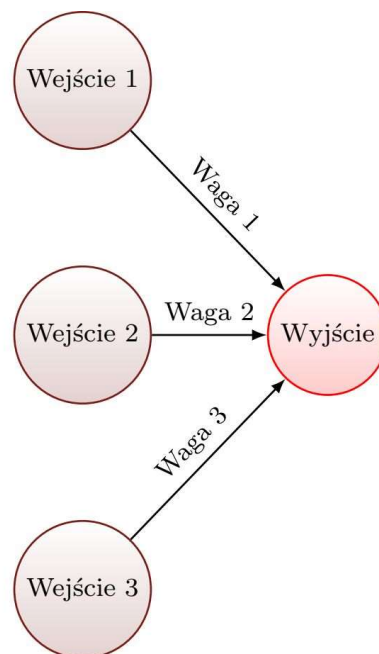
Zaimplementuj prostą sieć neuronową składającą się z jednego wejścia oraz jednego wyjścia:



Wartość podawana na wejściu jest mnożona przez wagę i zwracana przez sieć.

Zadanie 2:

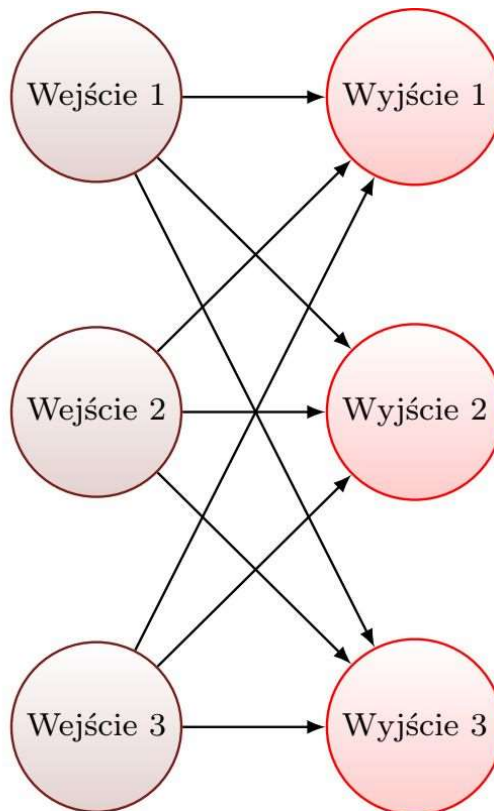
Zaimplementuj prostą sieć neuronową składającą się z trzech wejść oraz jednego wyjścia:



Wartości podawane na poszczególnych wejściach mnożone są przez wartość odpowiadającej im wagi, a następnie zwracane przez sieć.

Zadanie 3:

Zaimplementuj prostą sieć neuronową składającą się z trzech wejść oraz trzech wyjść:



Dla każdego neuronu wyjściowego wartości podawane na poszczególnych wejściach mnożone są przez wartość odpowiadającą im wagi. Przygotuj funkcję `neural_network`, która przyjmie wektor wartości wejściowych (w tym zadaniu będzie on 3-elementowy, ale przygotuj funkcję uniwersalną, która będzie działała dla dowolnie dużych wektorów) oraz macierz dwuwymiarową wag (liczba wierszy macierzy musi odpowiadać liczbie neuronów wyjściowych sieci, a liczba kolumn liczbie neuronów wejściowych). Funkcja powinna zwrócić wektor z wartościami poszczególnych neuronów warstwy wyjściowej. Zwróć uwagę, że liczba elementów w wektorze wejściowym musi odpowiadać liczbie kolumn w macierzy, a liczba elementów w wektorze wyjściowym liczbie wierszy w macierzy wag.

Przetestuj działanie sieci dla następujących danych:

Wagi:

$$weights = \begin{bmatrix} 0.1 & 0.1 & -0.3 \\ 0.1 & 0.2 & 0.0 \\ 0.0 & 1.3 & 0.1 \end{bmatrix}$$

Dane wejściowe (cztery serie danych):

$$input1 = [8.5 \quad 9.5 \quad 9.9 \quad 9.0]$$

$$input2 = [0.65 \quad 0.8 \quad 0.8 \quad 0.9]$$

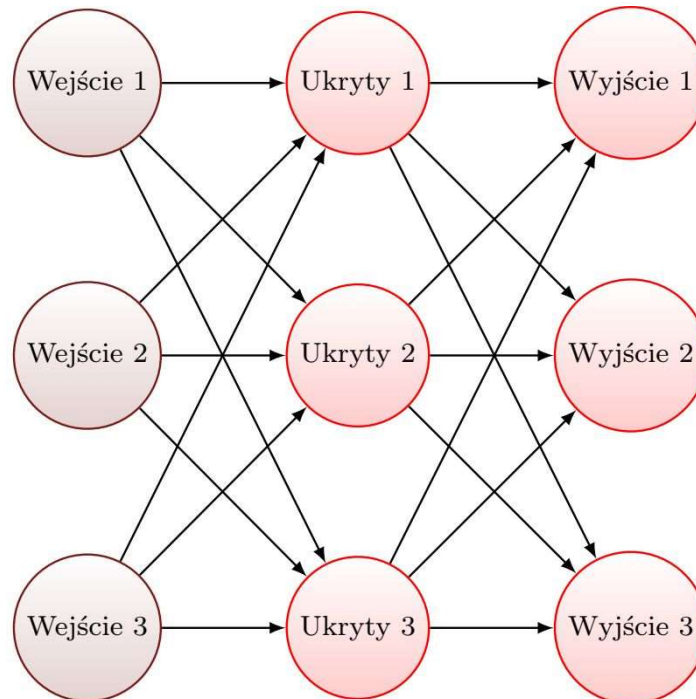
$$input3 = [1.2 \quad 1.3 \quad 0.5 \quad 1.0]$$

Oczekiwany wynik dla pierwszej serii:

$$output = [0.555 \quad 0.98 \quad 0.965]$$

Zadanie 4:

Zaimplementuj głęboką sieć neuronową składającą się z trzech wejść, trzech neuronów w warstwie ukrytej oraz trzech wyjść:



Każdy neuron znajdujący się w warstwie ukrytej (środkowa) oraz wyjściowej posiada zestaw wag, których liczba odpowiada liczbie neuronów w warstwie bezpośrednią go poprzedzającej. Obliczenie odpowiedzi takiej sieci dla danych wejściowych odbywa się najpierw poprzez obliczenie wartości neuronów w warstwie ukrytej (analogicznie do zadania 3), a następnie na podstawie tych wartości wyznaczeniu wartości neuronów w warstwie wyjściowej. Przygotuj funkcję `deep_neural_network`, która przyjmie wektor wartości wejściowych (w tym zadaniu będzie on 3-elementowy, ale przygotuj funkcję uniwersalną, która będzie działała dla dowolnie dużych wektorów) oraz macierz trzymiwarową wag (wagi dla każdego neuronu w każdej warstwie). Funkcja powinna zwrócić wektor z wartościami poszczególnych neuronów warstwy wyjściowej. Zwróć uwagę, że liczba neuronów w każdej z warstw może być różna.

Przetestuj działanie sieci dla następujących danych:

Wagi:

$$hidden = \begin{bmatrix} 0.1 & 0.2 & -0.1 \\ -0.1 & 0.1 & 0.9 \\ 0.1 & 0.4 & 0.1 \end{bmatrix}$$

$$output = \begin{bmatrix} 0.3 & 1.1 & -0.3 \\ 0.1 & 0.2 & 0.0 \\ 0.0 & 1.3 & 0.1 \end{bmatrix}$$

$$weights = [input \quad output]$$

Dane wejściowe (cztery serie danych):

$input1 = [8.5 \ 9.5 \ 9.9 \ 9.0]$

$input2 = [0.65 \ 0.8 \ 0.8 \ 0.9]$

$input3 = [1.2 \ 1.3 \ 0.5 \ 1.0]$

Zadanie 5:

Przygotuj program, który pozwoli na budowanie sieci składającej się z warstw w pełni połączonych (ang. *fully connected layers*), czyli takich jak w poprzednich zadaniach. Program ma umożliwiać budowanie sieci o dowolnej liczbie neuronów wejściowych oraz wyjściowych, dowolnej liczbie warstw ukrytych z dowolną liczbą neuronów. Program powinien na początek generować losowe wartości wag dla każdego z neuronów.