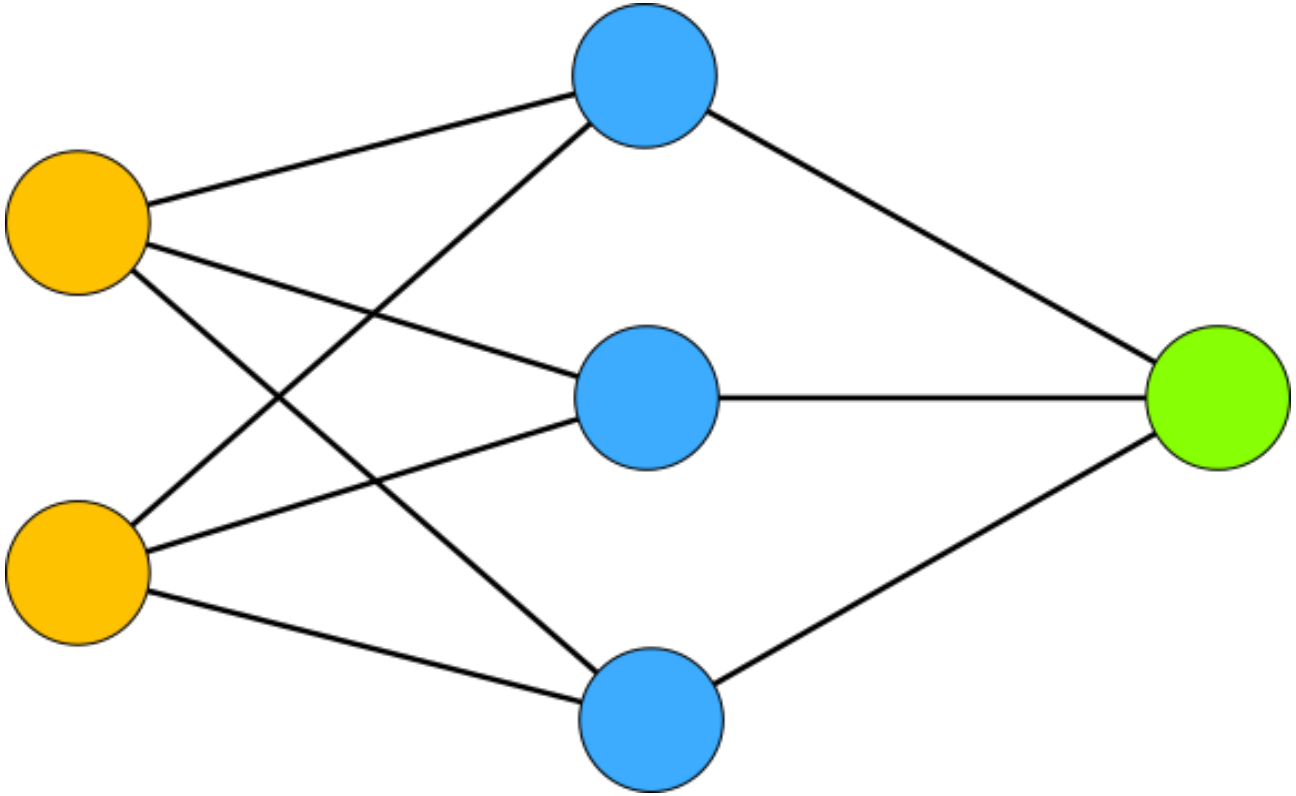
Sieci Neuronowe

Plan warsztatu

1. Czym jest sieć neuronowa
2. Tworzenie sieci typu Feed-forward
3. Uczenie sieci neuronowej czyli funkcja straty, optymalizacja
4. Back-propagation (propagacja wsteczna)
5. Uczenie/Trenowanie sieci neuronowej

Czym jest sieć neuronowa?

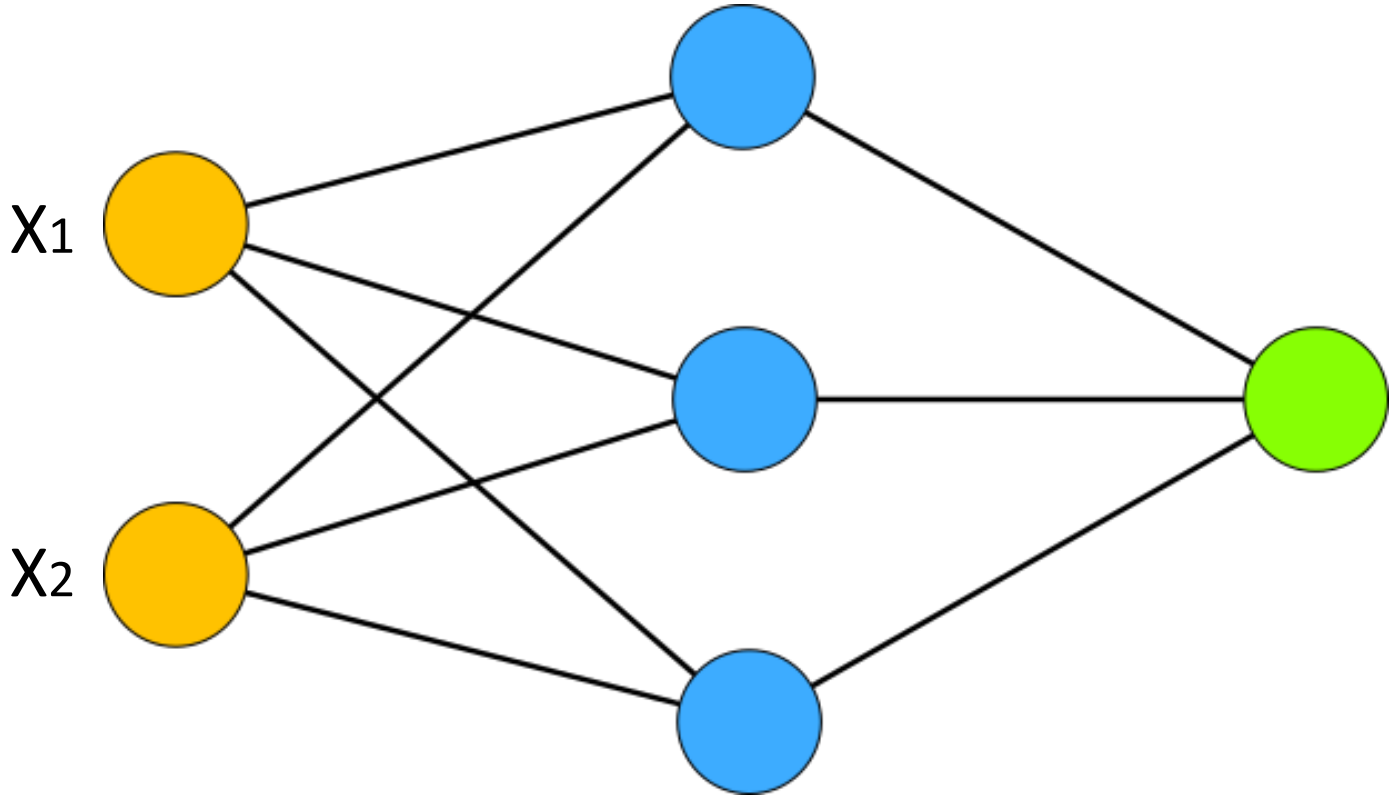
Zbiór neuronów i połączeń między nimi



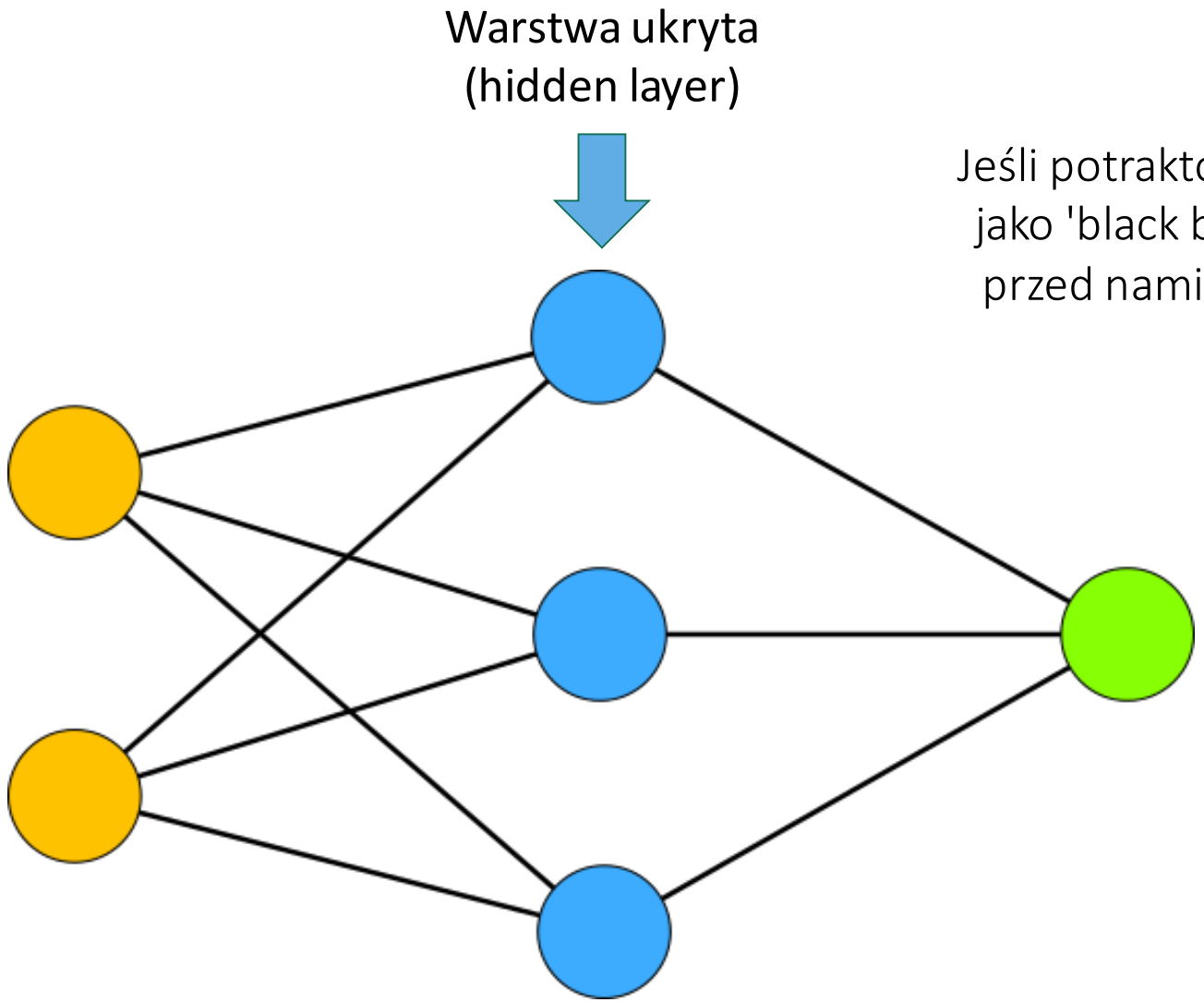
Czym jest sieć neuronowa? - warstwy

Wejście (input)

Oznaczane zwykle jako X, gdzie X jest wektorem wszystkich wartości, tzw. cech (ang. features)

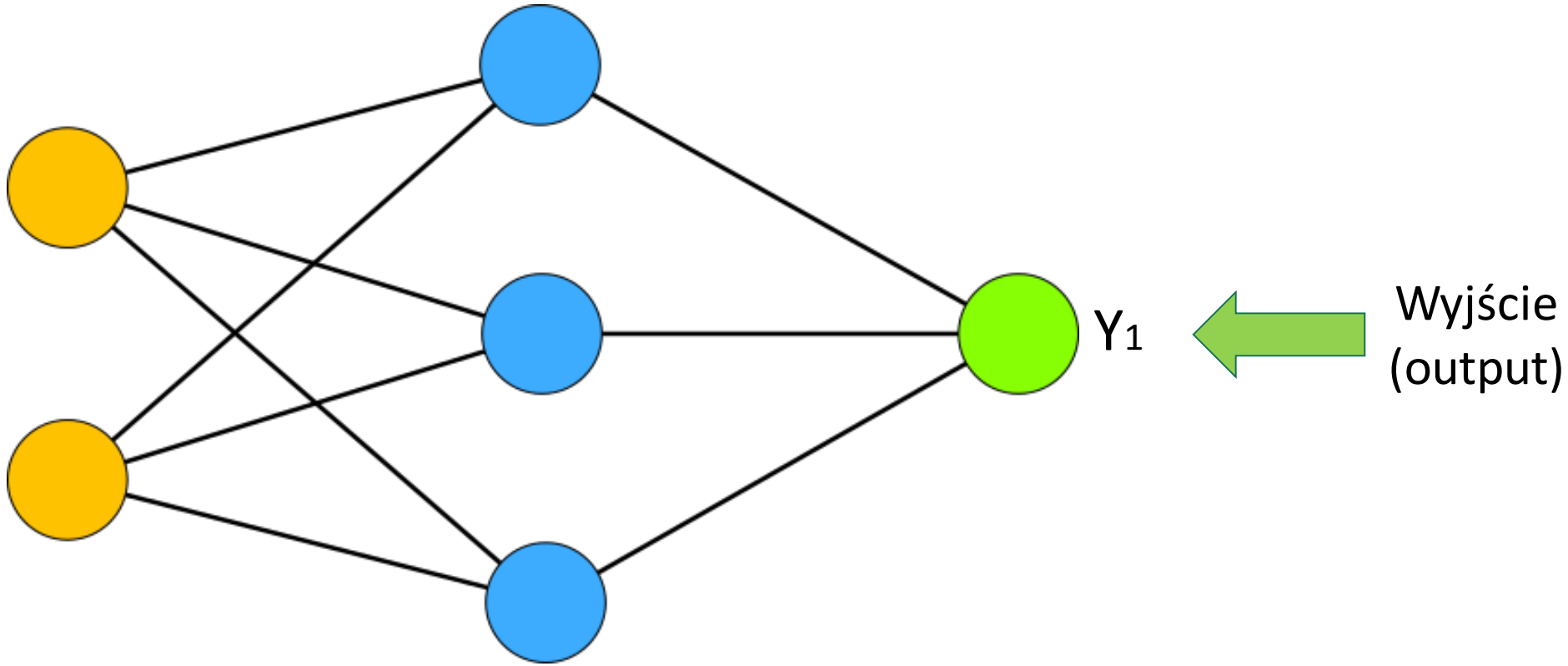


Czym jest sieć neuronowa? - warstwy

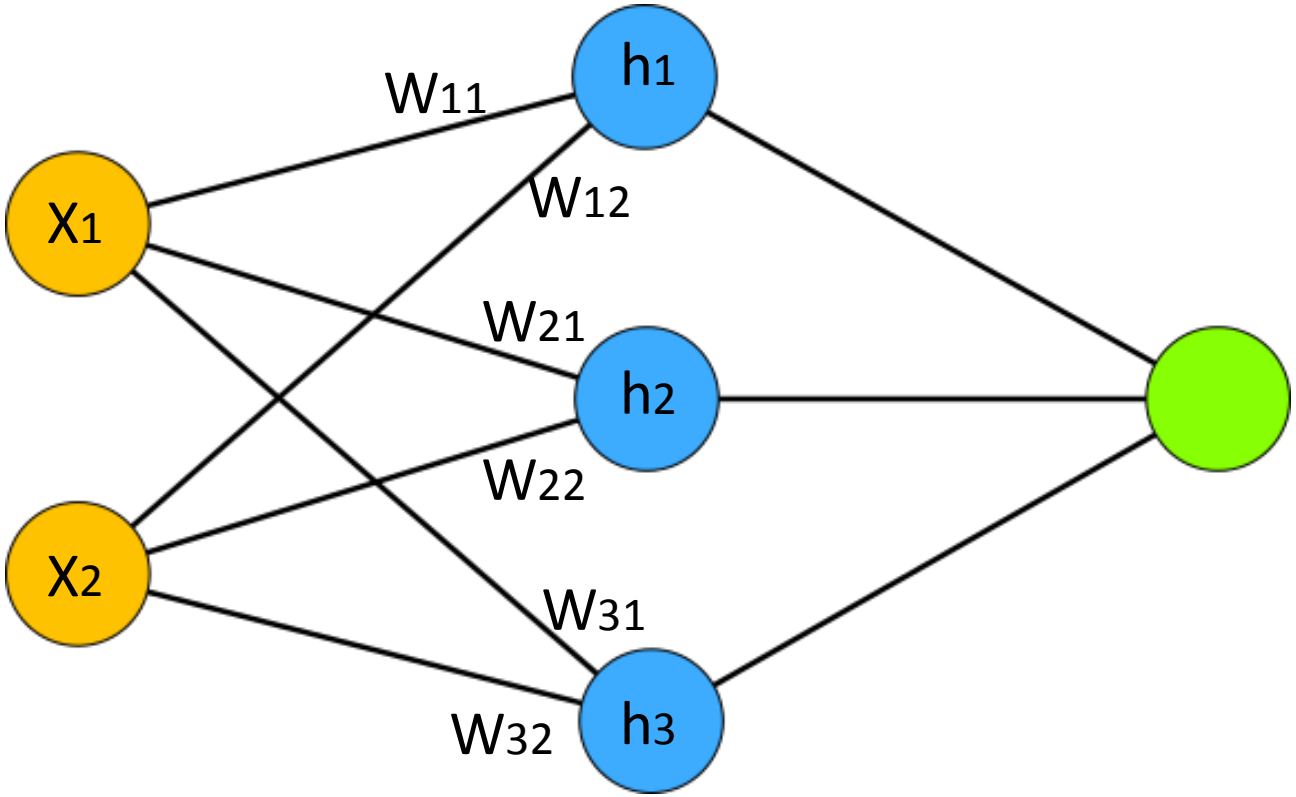


Jeśli potraktować sieć neuronową jako 'black box' to te warstwy są przed nami ukryte, stąd nazwa.

Czym jest sieć neuronowa? - warstwy

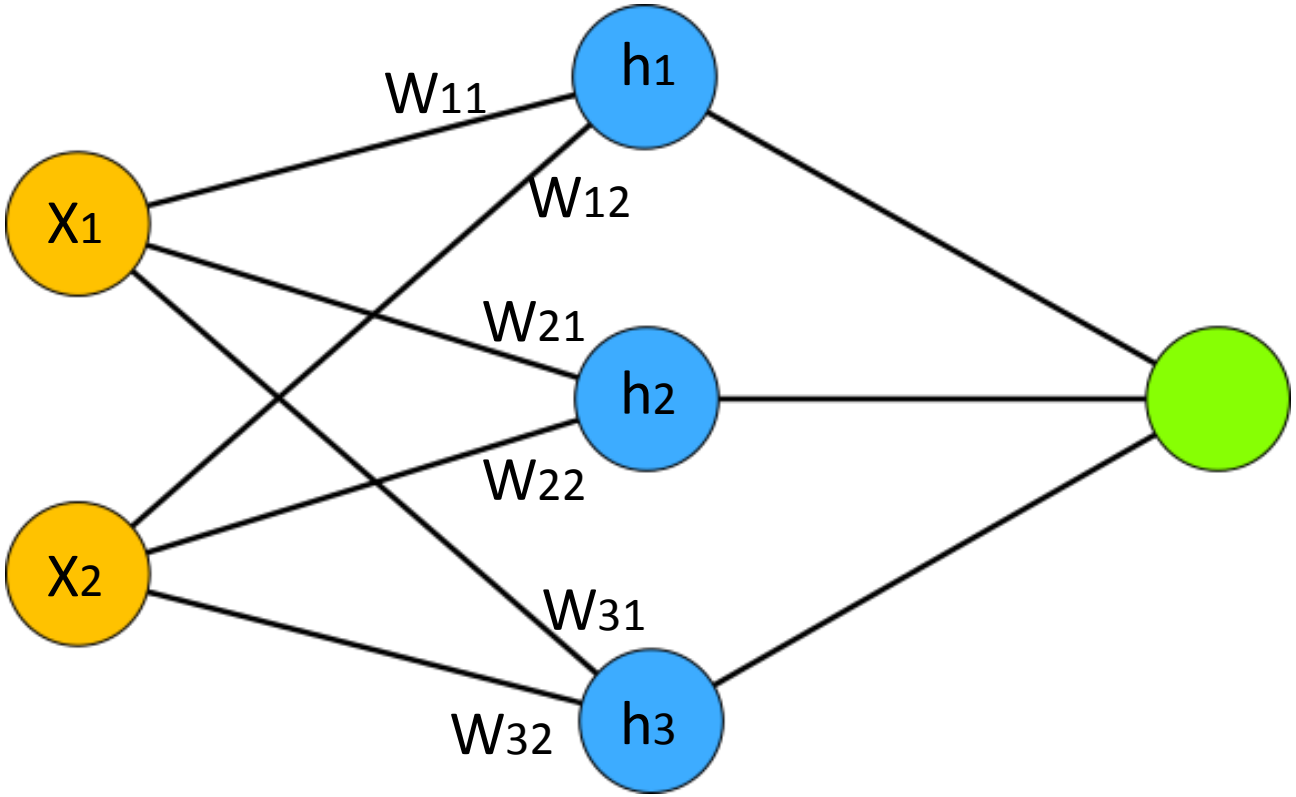


Czym jest sieć neuronowa? - połączenia



Każde połączenie ma przypisaną wagę. Tutaj oznaczoną W11, W12, itd.

Czym jest sieć neuronowa? - neuron



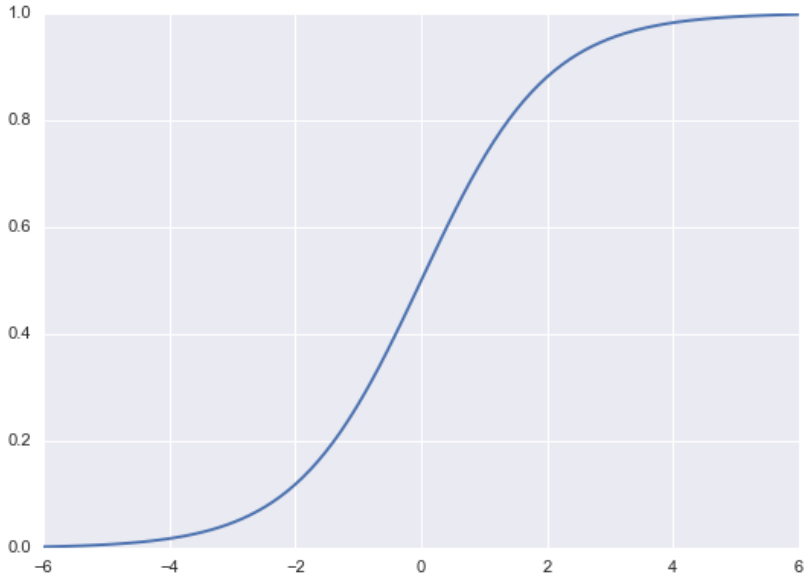
Każde połączenie ma przypisaną wagę. Tutaj oznaczoną W11, W12, itd.

Neuron wylicza ważoną sumę wejść i aplikuje funkcję aktywacji.

$$h_1 = f\left(\sum_i x_i \cdot w_{1i}\right)$$

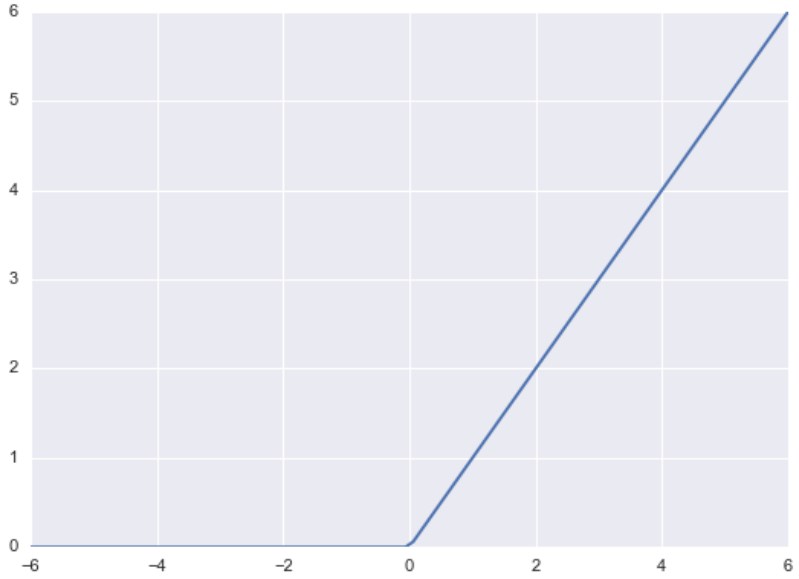
Funkcja aktywacji

Czym jest sieć neuronowa? - przykładowe funkcje aktywacji



Sigmoid

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



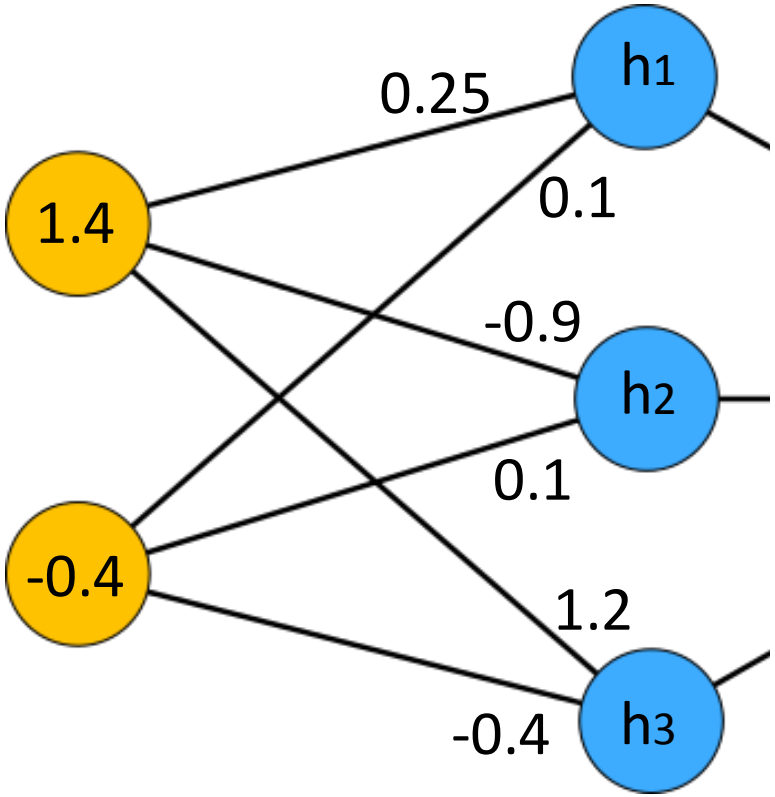
ReLU

$$ReLU(x) = max(0, x)$$

Z tej funkcji będziemy później korzystali przy implementacji sieci neuronowej.

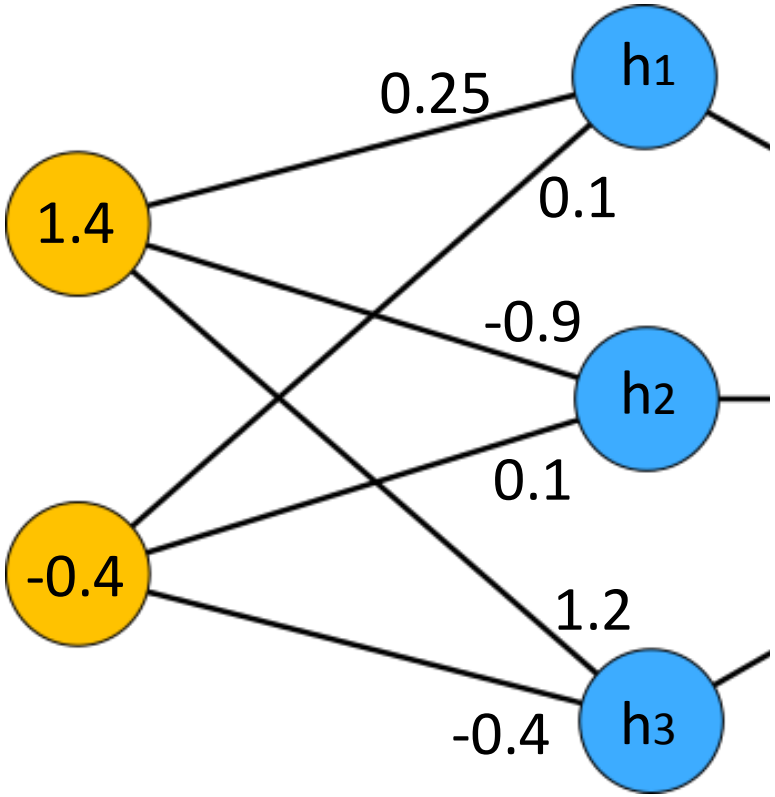
Czym jest sieć neuronowa? - przykład

Policzmy jakie otrzymamy wartości dla warstwy ukrytej i aktywacji sigmoidalnej.



Czym jest sieć neuronowa? - przykład

Policzmy jakie otrzymamy wartości dla warstwy ukrytej i aktywacji sigmoidalnej.

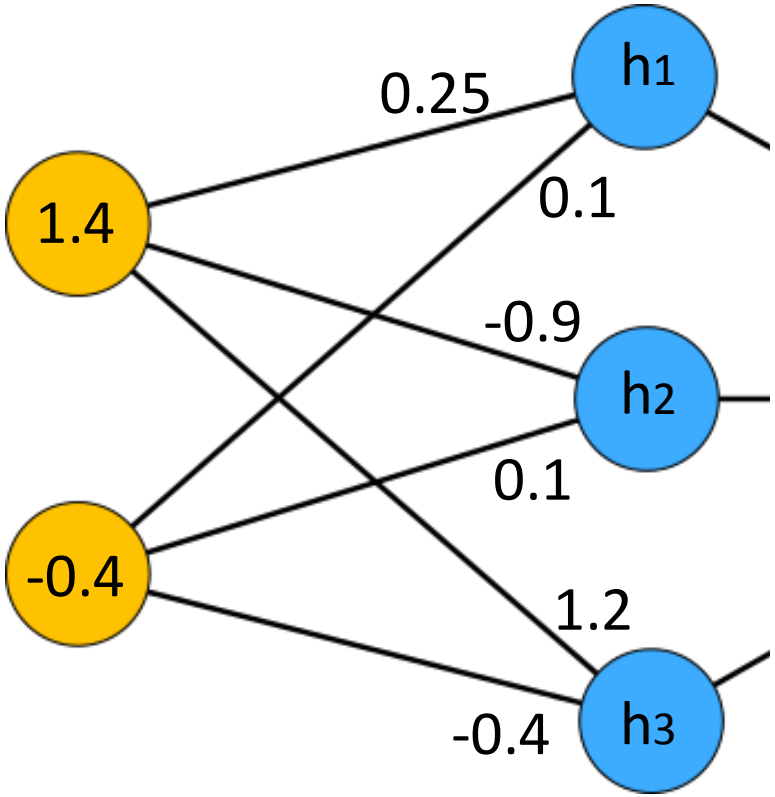


$$z_1 = 1.4 \cdot 0.25 + (-0.4) \cdot 0.1$$

$$h_1 = \sigma(z_1)$$

Czym jest sieć neuronowa? - przykład

Policzmy jakie otrzymamy wartości dla warstwy ukrytej i aktywacji sigmoidalnej.

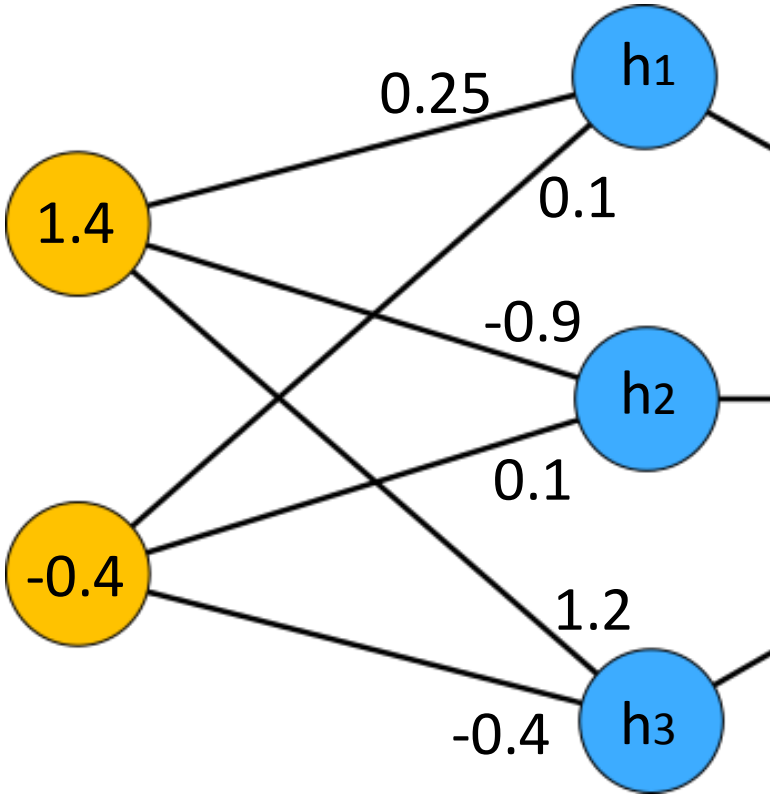


$$z_1 = 1.4 \cdot 0.25 + (-0.4) \cdot 0.1$$
$$h_1 = \sigma(z_1)$$

$$z_2 = 1.4 \cdot (-0.9) + (-0.4) \cdot 0.1$$
$$h_2 = \sigma(z_2)$$

Czym jest sieć neuronowa? - przykład

Policzmy jakie otrzymamy wartości dla warstwy ukrytej i aktywacji sigmoidalnej.



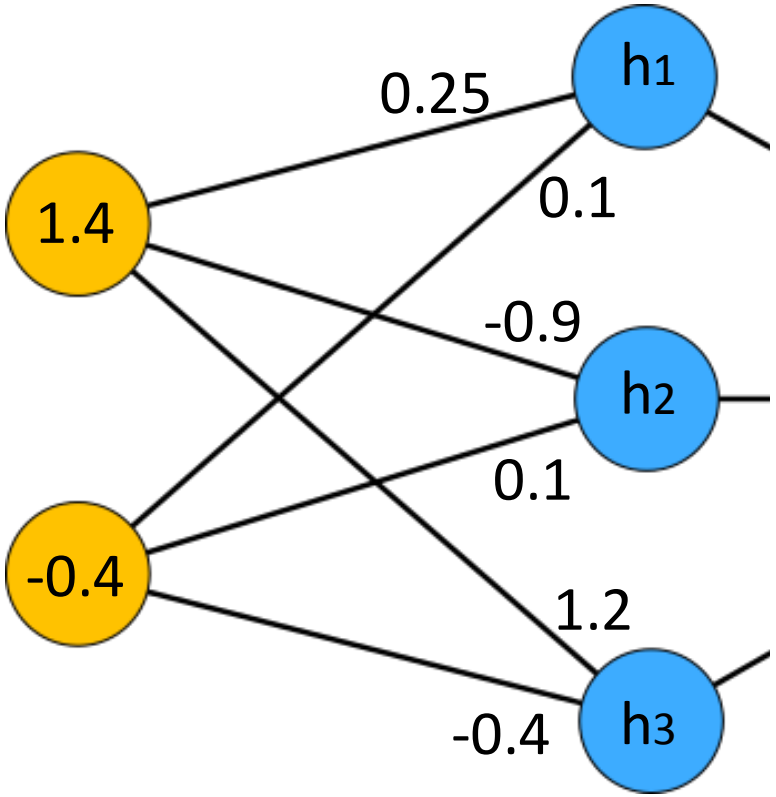
$$z_1 = 1.4 \cdot 0.25 + (-0.4) \cdot 0.1$$
$$h_1 = \sigma(z_1)$$

$$z_2 = 1.4 \cdot (-0.9) + (-0.4) \cdot 0.1$$
$$h_2 = \sigma(z_2)$$

$$z_3 = 1.4 \cdot 1.2 + (-0.4) \cdot (-0.4)$$
$$h_3 = \sigma(z_3)$$

Czym jest sieć neuronowa? - przykład

Całość można kompaktowo zapisać przy pomocy notacji macierzowej.



$$z = x \cdot W$$

$$z = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \cdot \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix}$$

$$h = \sigma(z)$$

Funkcje aktywacji aplikujemy per element. (ang. element-wise)

Sieć typu Feed-forward - kodzenie

Zaimplementujemy teraz tzw. Forward pass dla sieci neuronowej.

Szczegóły:

- Sieć będzie miała **jedną** warstwę ukrytą (niech każdy dobierze jej wielkość wg uznania)
- Użyjemy funkcji **sigmoid** jako funkcję aktywacji.

Uczenie sieci neuronowej – funkcja kosztu

By móc wytrenować sieć potrzebujemy w jakiś sposób określić jak dobrze sobie radzi z postawionym przez nas zadaniem.

Uczenie sieci neuronowej – funkcja kosztu

By móc wytrenować sieć potrzebujemy w jakiś sposób określić jak dobrze sobie radzi z postawionym przez nas zadaniem.

Do tego właśnie wykorzystywane są funkcje straty (ang. loss function). Określają one w sposób matematyczny jak bardzo niepoprawna jest odpowiedź udzielona przez sieć neuronową.

Uczenie sieci neuronowej – funkcja straty

By móc wytrenować sieć potrzebujemy w jakiś sposób określić jak dobrze sobie radzi z postawionym przez nas zadaniem.

Do tego właśnie wykorzystywane są funkcje straty (ang. loss function). Określają one w sposób matematyczny jak bardzo niepoprawna jest odpowiedź udzielona przez sieć neuronową.

$$\mathcal{L}(y, \hat{y}) = \frac{1}{n}(y - \hat{y})^2$$

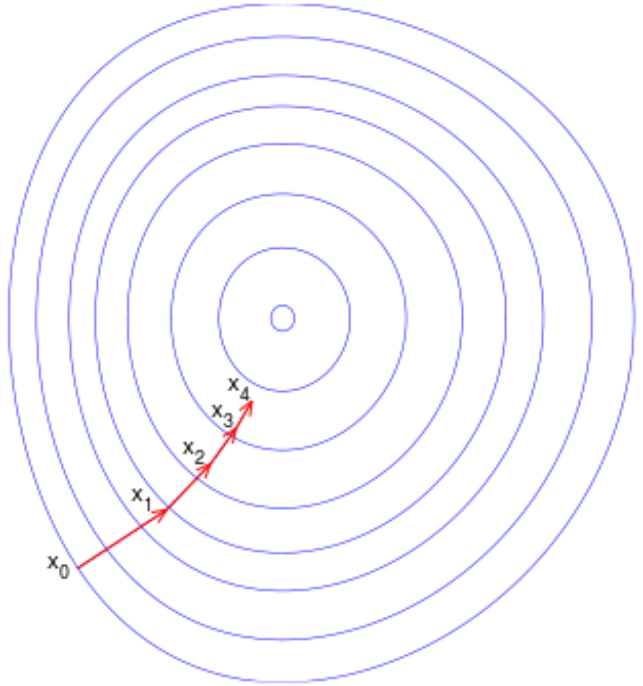
Przykładowa funkcja kosztu z której będziemy korzystać podczas implementacji uczenia.

Tzw. błąd średnio kwadratowy
(MSE – Mean Square Error)

Uczenie sieci neuronowej – optymalizacja

Mając już funkcję straty wiemy, że nasza sieć neuronowa będzie podawała lepsze odpowiedzi, kiedy ta funkcja będzie jak najmniejsza. Potrzebujemy zatem *minimalizować* tą funkcję.

Najczęściej wykorzystuje się do tego pewną odmianę algorytmu gradientowego, który polega na podążaniu w kierunku, który wyznacza nam gradient.



Uczenie sieci neuronowej – optymalizacja

No dobrze ale skąd weźmiemy gradient?

Uczenie sieci neuronowej – optymalizacja

No dobrze ale skąd weźmiemy gradient?

Do wyliczania gradientów wykorzystuje się tzw. Back-propagation (propagacja wsteczna). Metoda ta pozwala w wydajny sposób wyliczyć gradienty dla wszystkich parametrów.

Bazuje ona praktycznie w całości na Regule Łańcuchowej (ang. Chain Rule).

$$\frac{df}{dx} = \frac{df}{dg} \cdot \frac{dg}{dx}$$