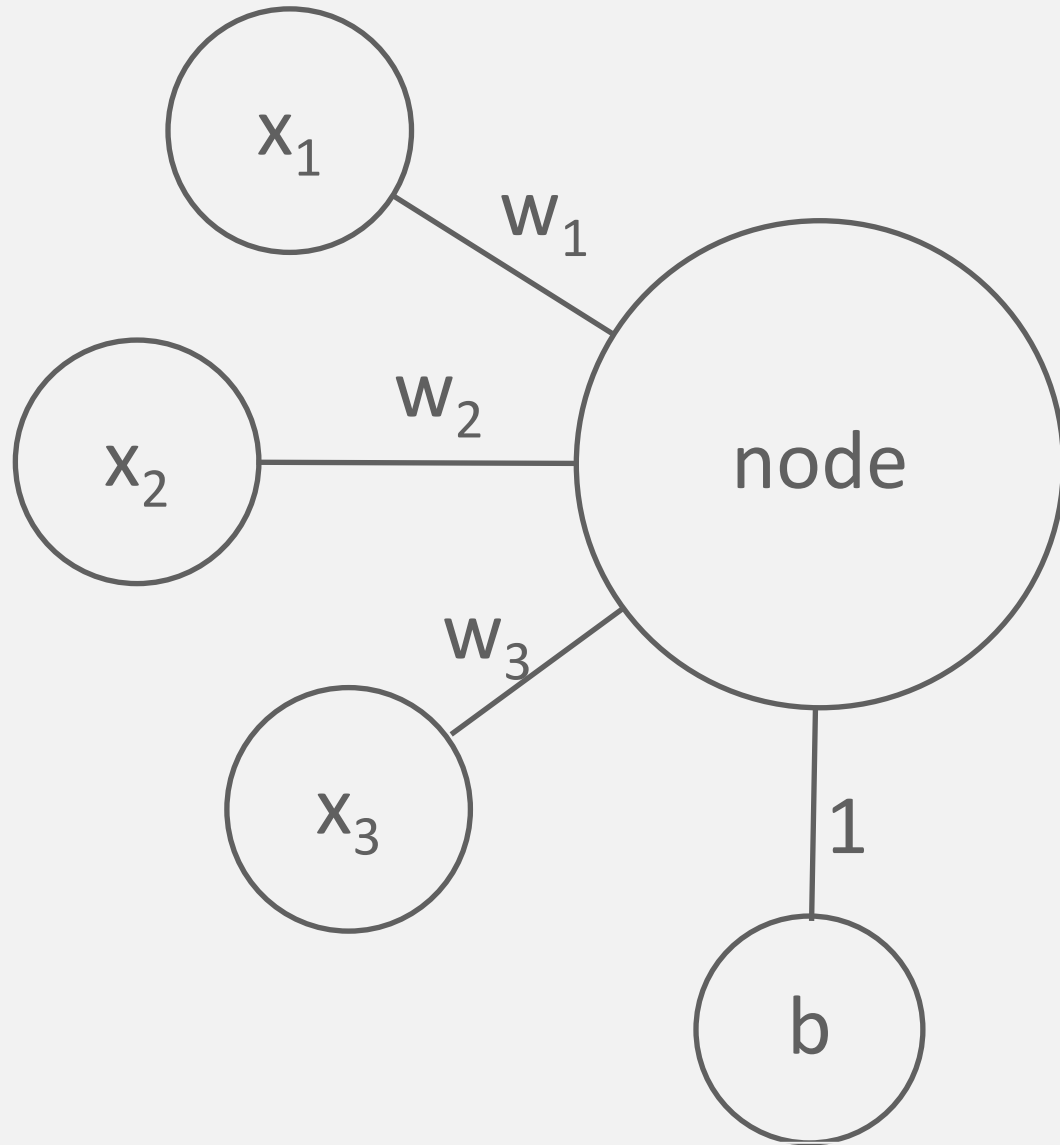


analiza danych

1. Wykład wstępny
2. Elementy algebry liniowej
3. Wybrane aspekty obliczeń komputerowych
- 4. Architektura pojedynczej sieci neuronowej cz.1**
5. Architektura pojedynczej sieci neuronowej cz. 2
6. Architektura modelu uczenia głębokiego - wprowadzenie
7. Architektura modelu uczenia głębokiego – rozszerzenia
8. Walidacja modelu
9. Przykłady innych struktur sieci
10. Przykłady własnych badań cz. 1
11. Przykłady własnych badań cz. 2
12. -15. Prezentacje zaliczeniowe

-
1. Pojedynczy neuron
 2. Przykładowe funkcje aktywacji
 3. Pojedyncza sieć neuronowa
 4. Zestaw sieci neuronowych
 5. Architektura modelu uczenia głębokiego

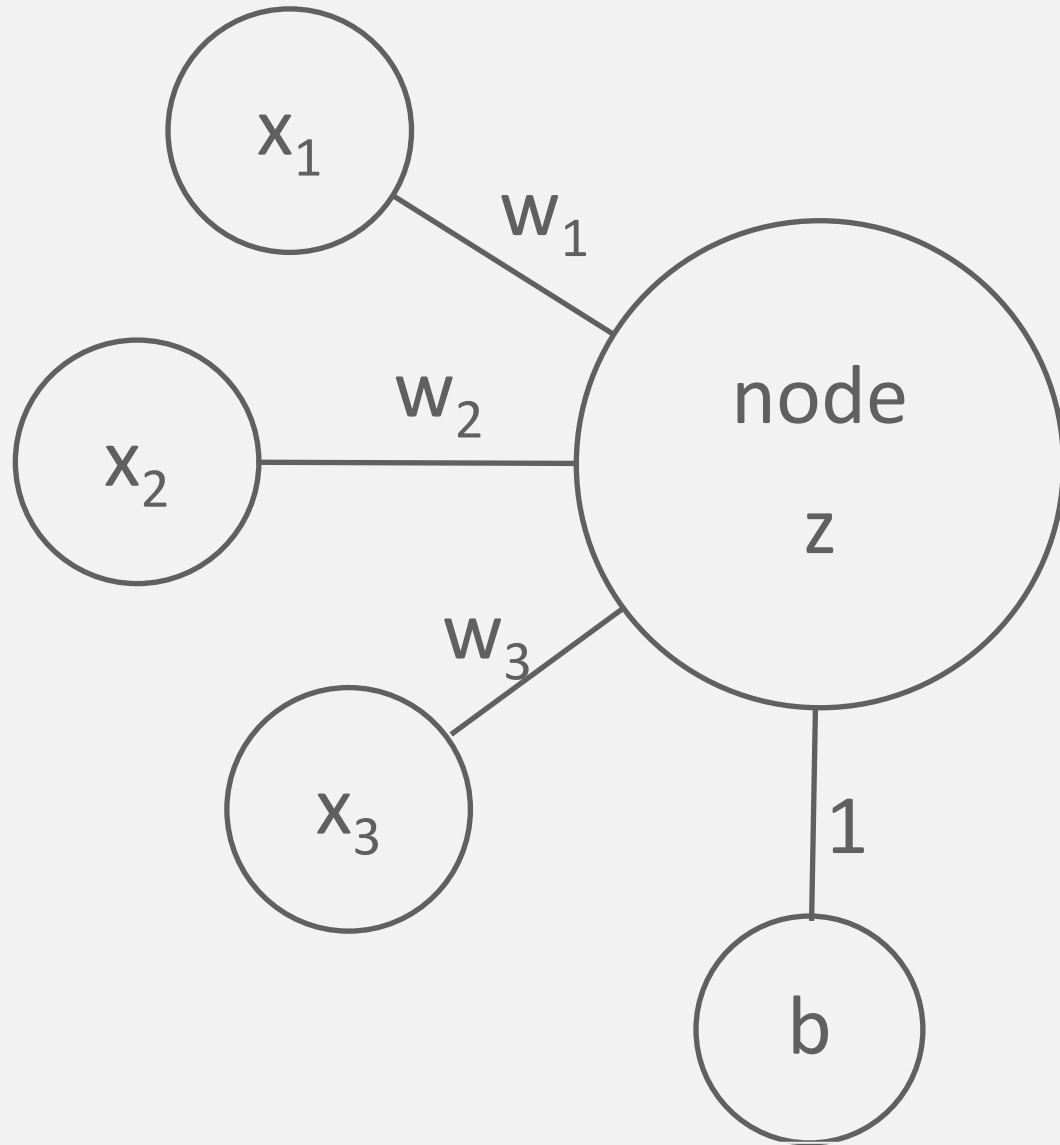
Pojedynczy neuron \rightarrow perceptron



Dane wejściowe

- x_i wartości wejściowe
 - oryginalne dane
 - wartości z poprzedniego neuronu
- w_2 wagi
- b bias / intercept / wyraz wolny

Pojedynczy neuron

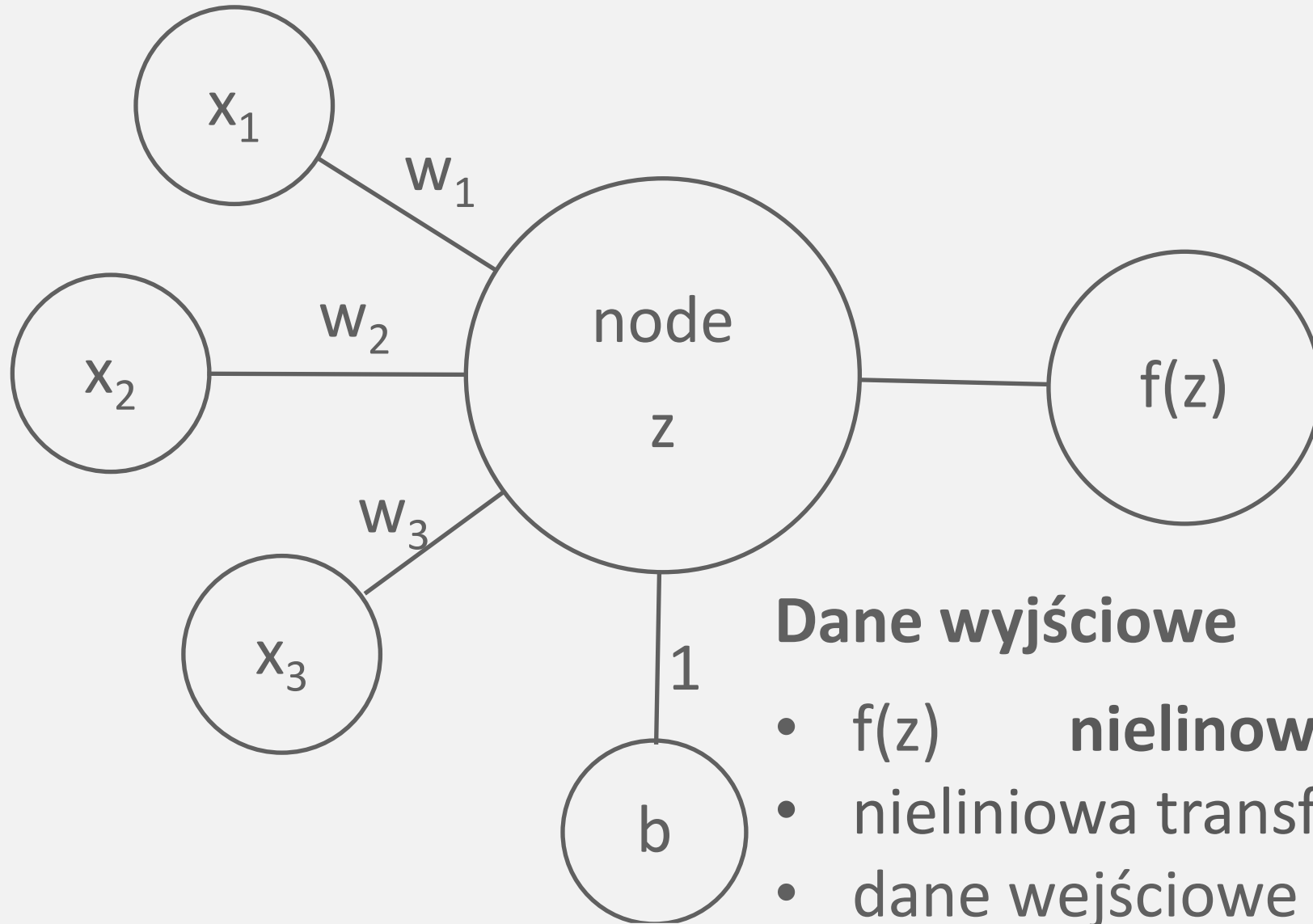


Wartość neuronu

Równanie regresji liniowej:

$$z = \hat{b} + x_1 \hat{w}_1 + x_2 \hat{w}_2 + x_3 \hat{w}_3$$

Pojedynczy neuron



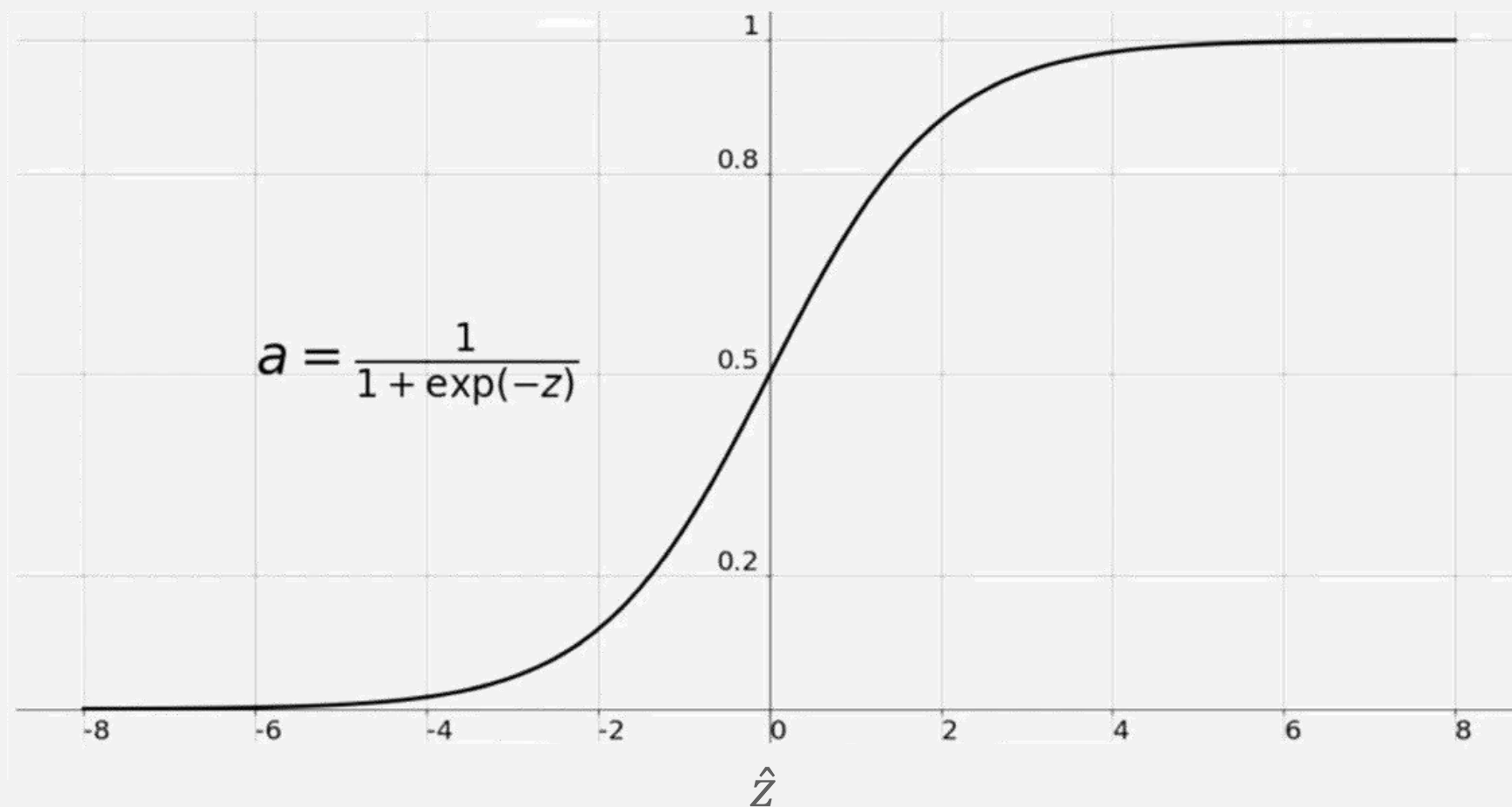
Dane wyjściowe

- $f(z)$ **nielinowa** funkcja aktywacji
- nielinowa transformacja danych
- dane wejściowe do następnego neuronu

Przykładowe funkcje aktywacji

sigmoid: $f(z) = \frac{1}{1+e^{\hat{z}}}$

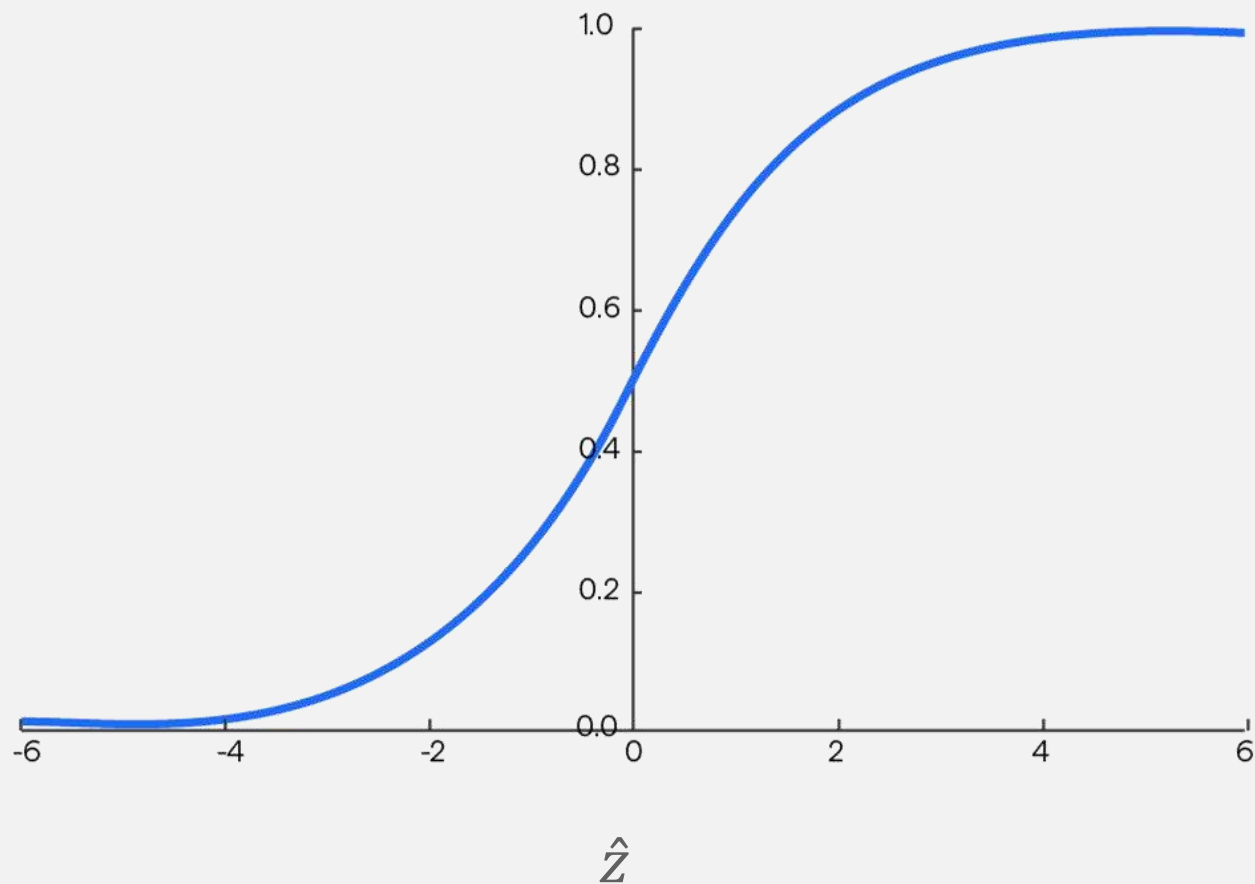
$f(z)$



Przykładowe funkcje aktywacji

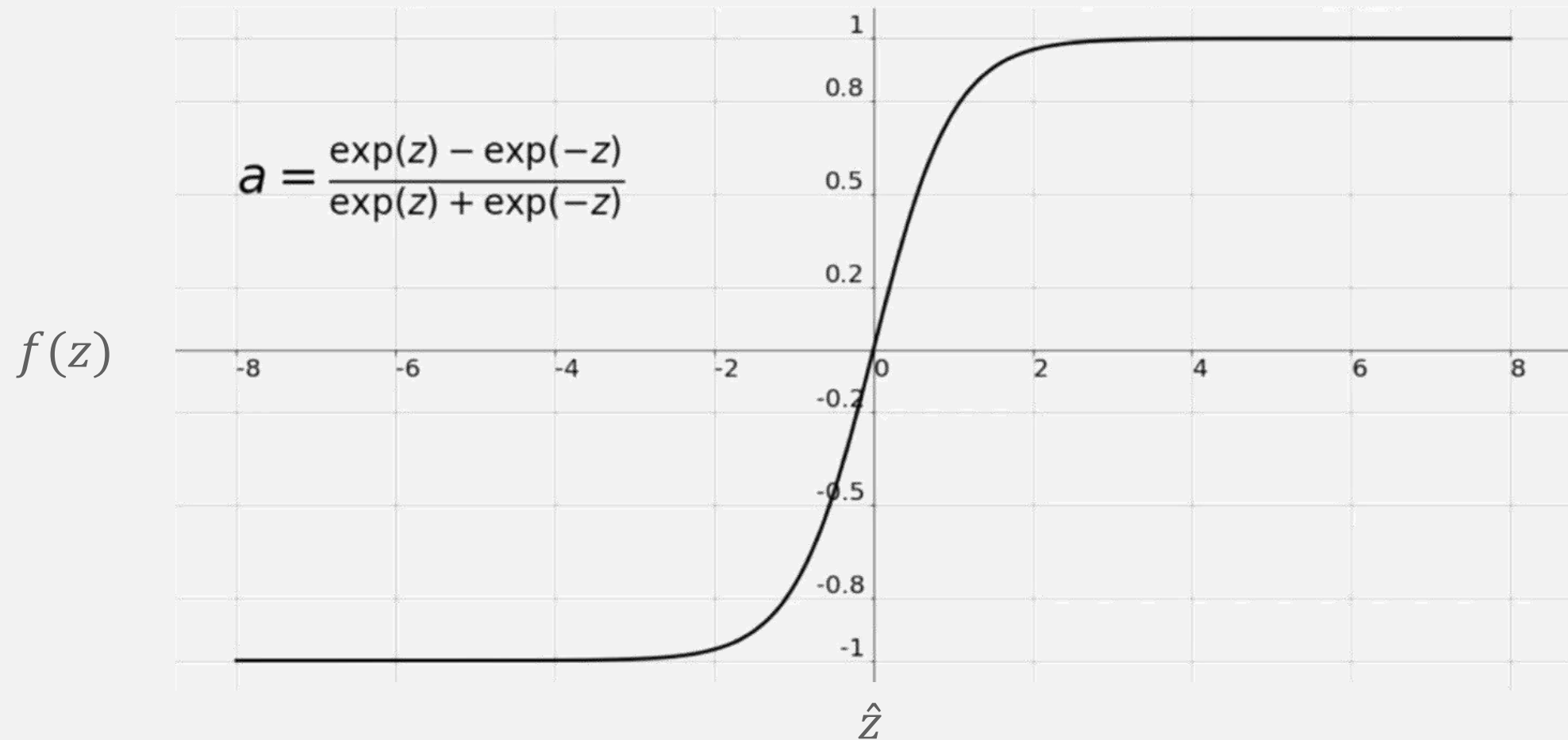
$$\text{softmax: } f(z) = \frac{e^{\hat{z}_i}}{\sum_{i=1}^K e^{\hat{z}_i}}$$

$f(z)$



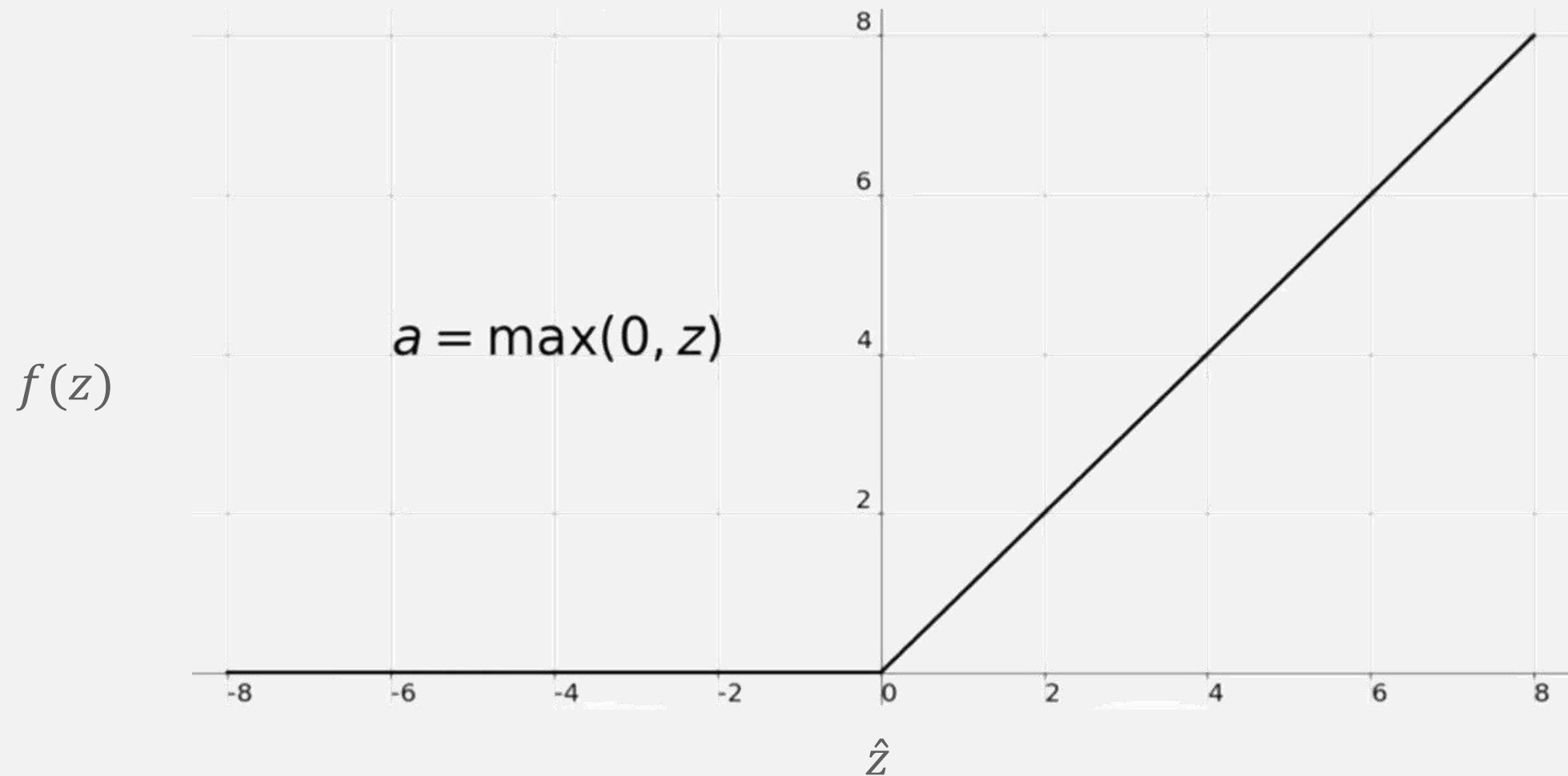
Przykładowe funkcje aktywacji

Hyperbolic Tangent Function / tanh: $f(z) = \frac{e^{\hat{z}} - e^{-\hat{z}}}{e^{\hat{z}} + e^{-\hat{z}}}$



Przykładowe funkcje aktywacji

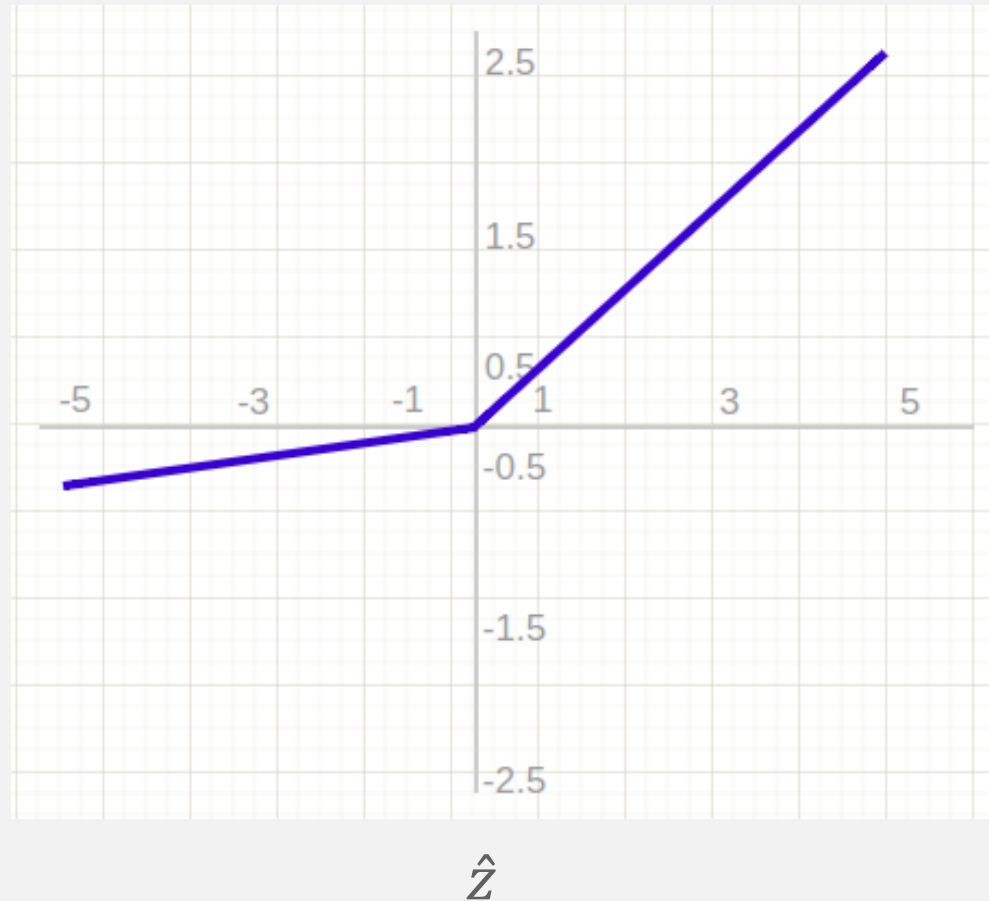
rectified linear unit / ReLU: $f(z) = \begin{cases} 0 & \hat{z} < 0 \\ z & \hat{z} \geq 0 \end{cases}$



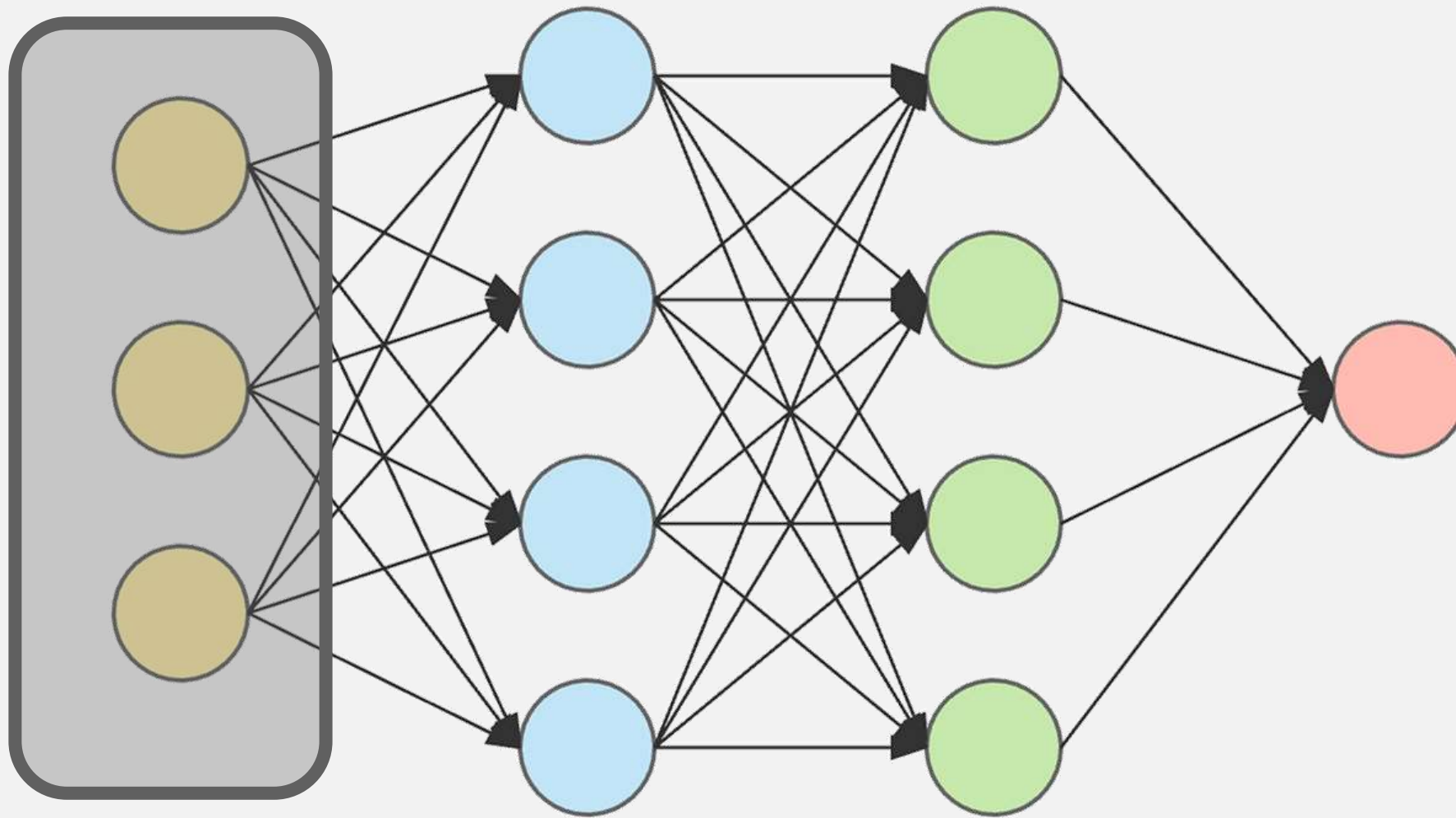
Przykładowe funkcje aktywacji

Leaky rectified linear unit / LReLU: $f(z) = \begin{cases} \alpha z & \hat{z} < 0 \\ z & \hat{z} \geq 0 \end{cases}$

$f(z)$



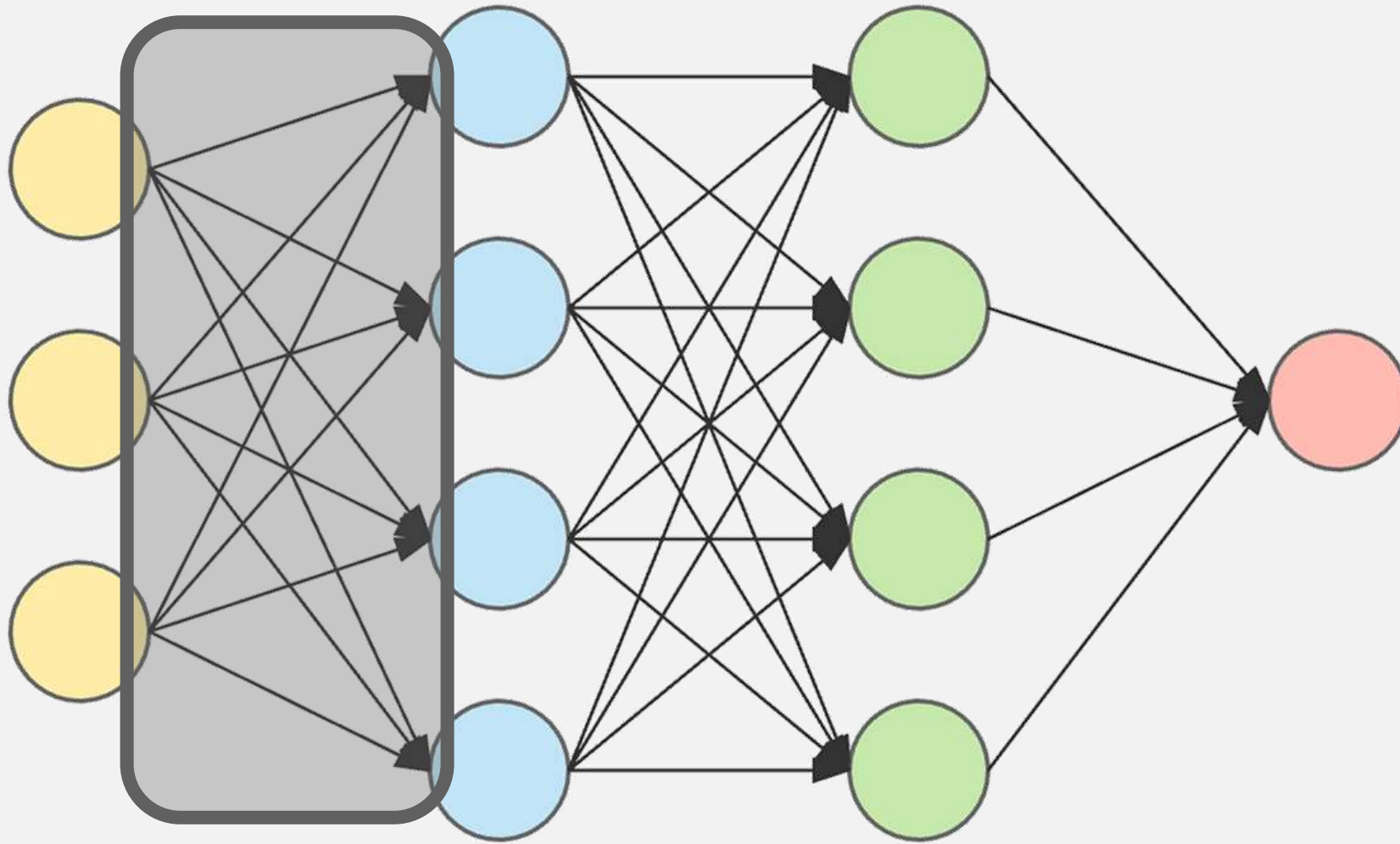
Wiele neuronów → sieć neuronowa



Warstwa wejściowa

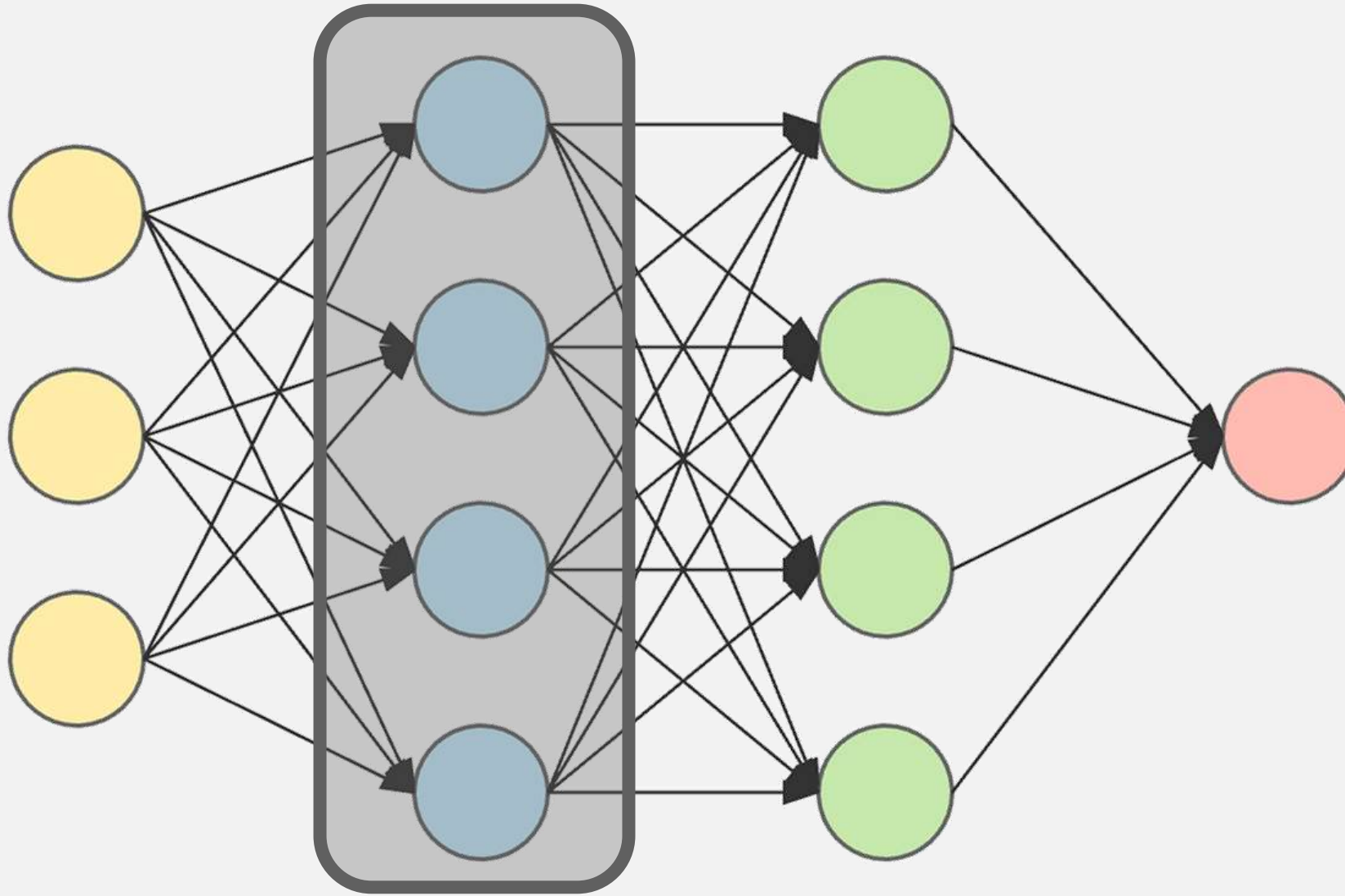
Oryginalne dane wejściowe / zmienne objaśniające / features : x_i

Sieć neuronowa



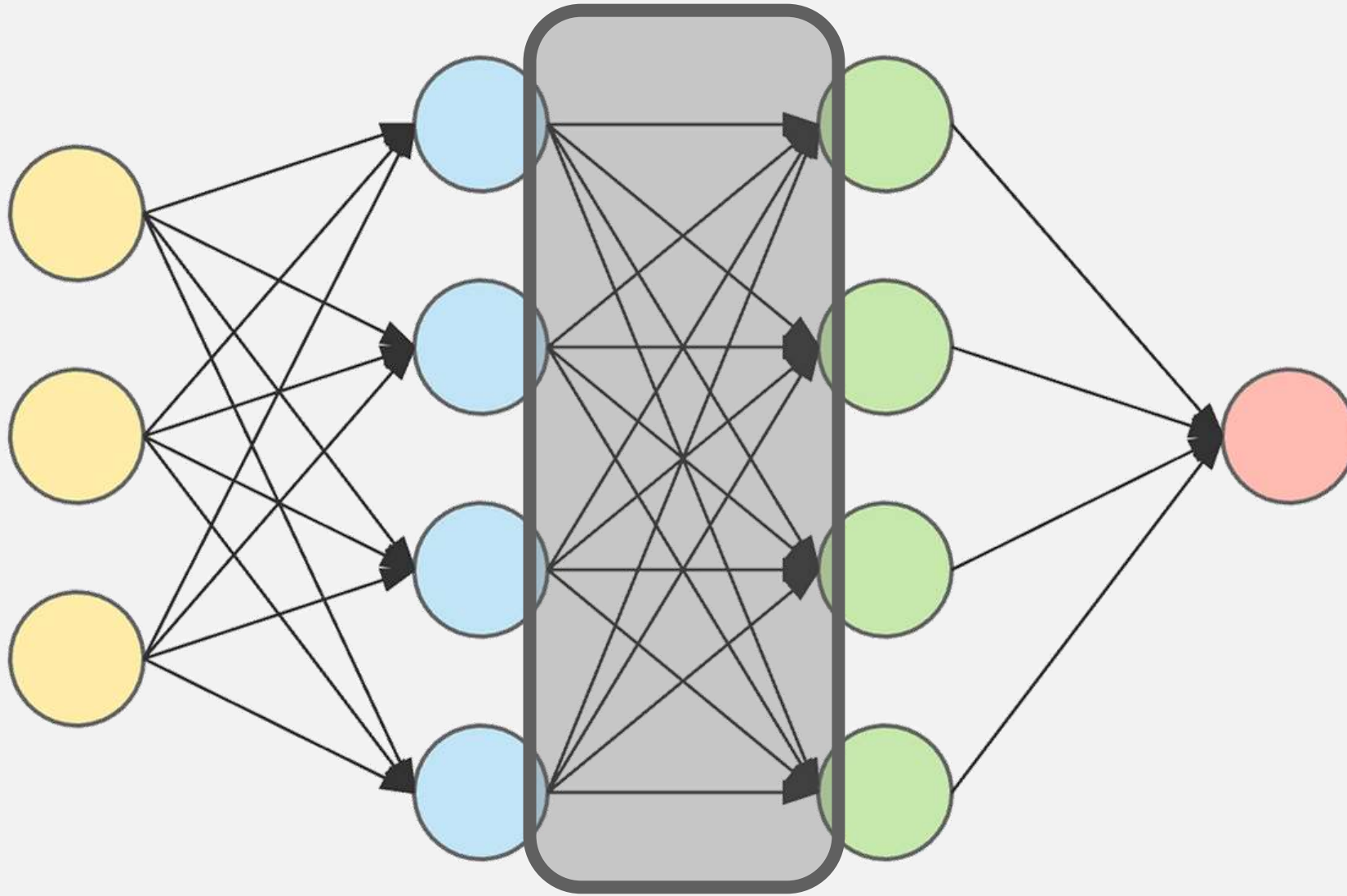
Wagi: w_i

Sieć neuronowa



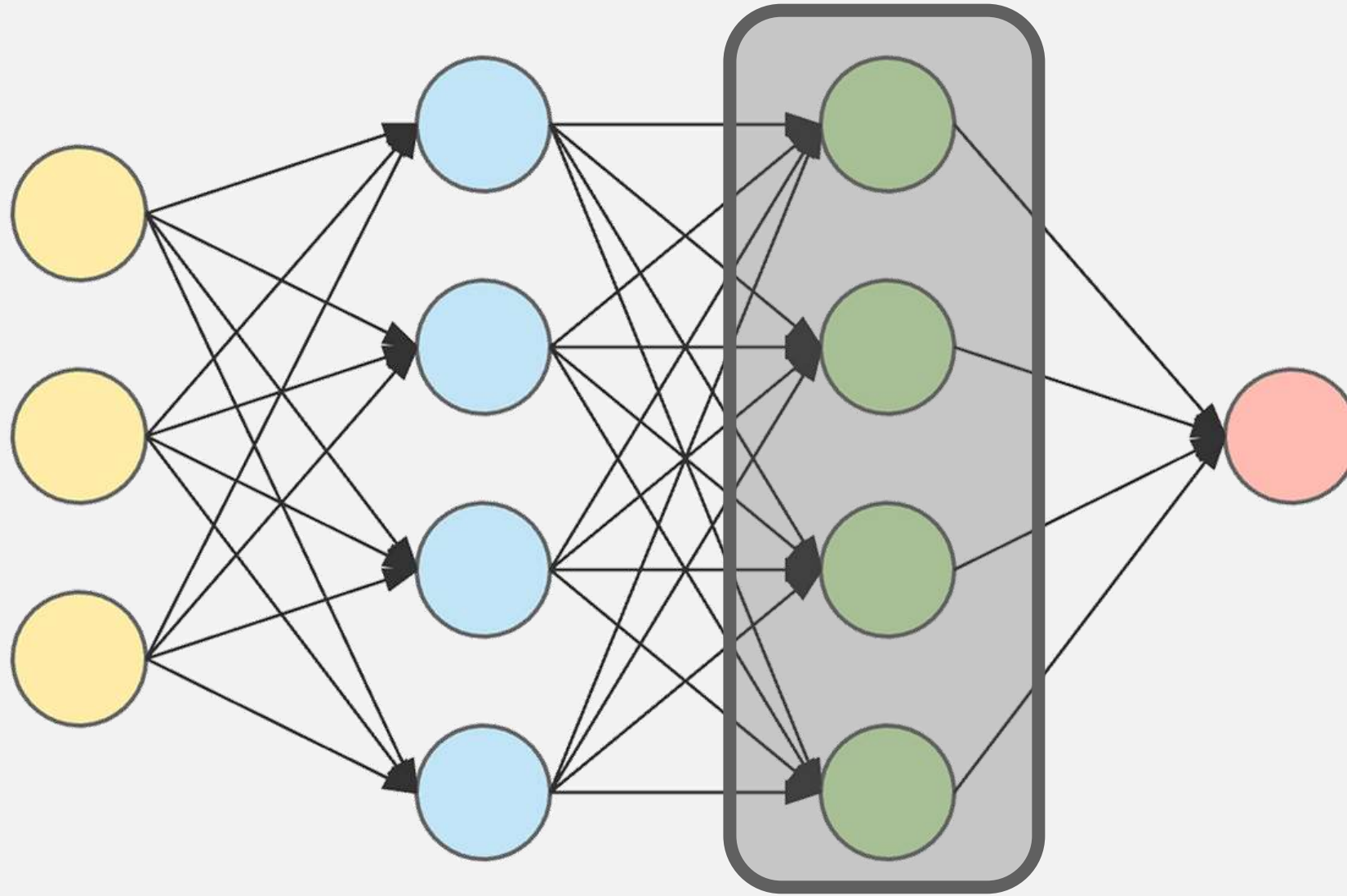
1. Warstwa neuronów / warstwa ukryta / Output 1. : $f(z_i)$

Sieć neuronowa



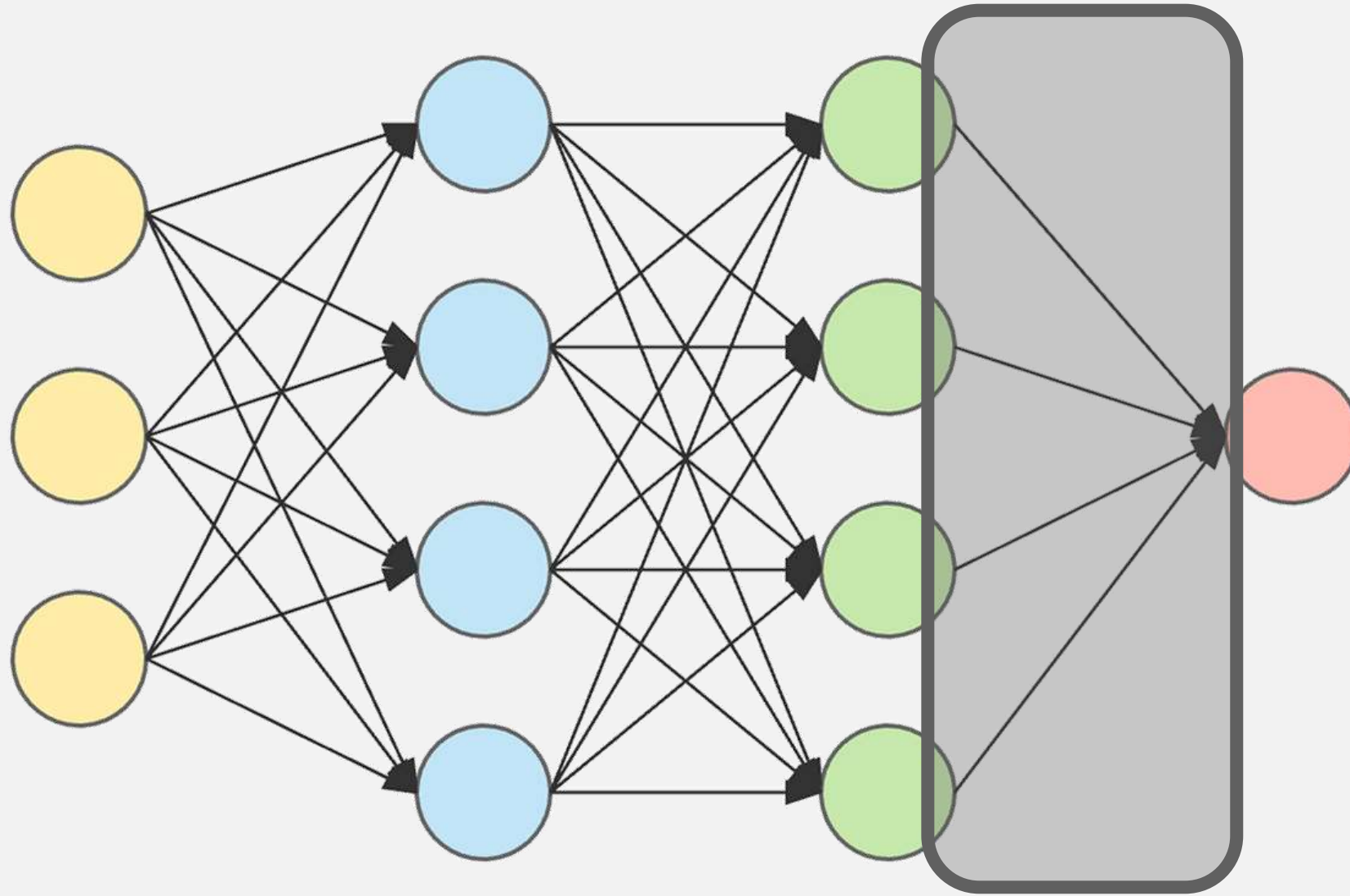
Wagi: w_i

Sieć neuronowa



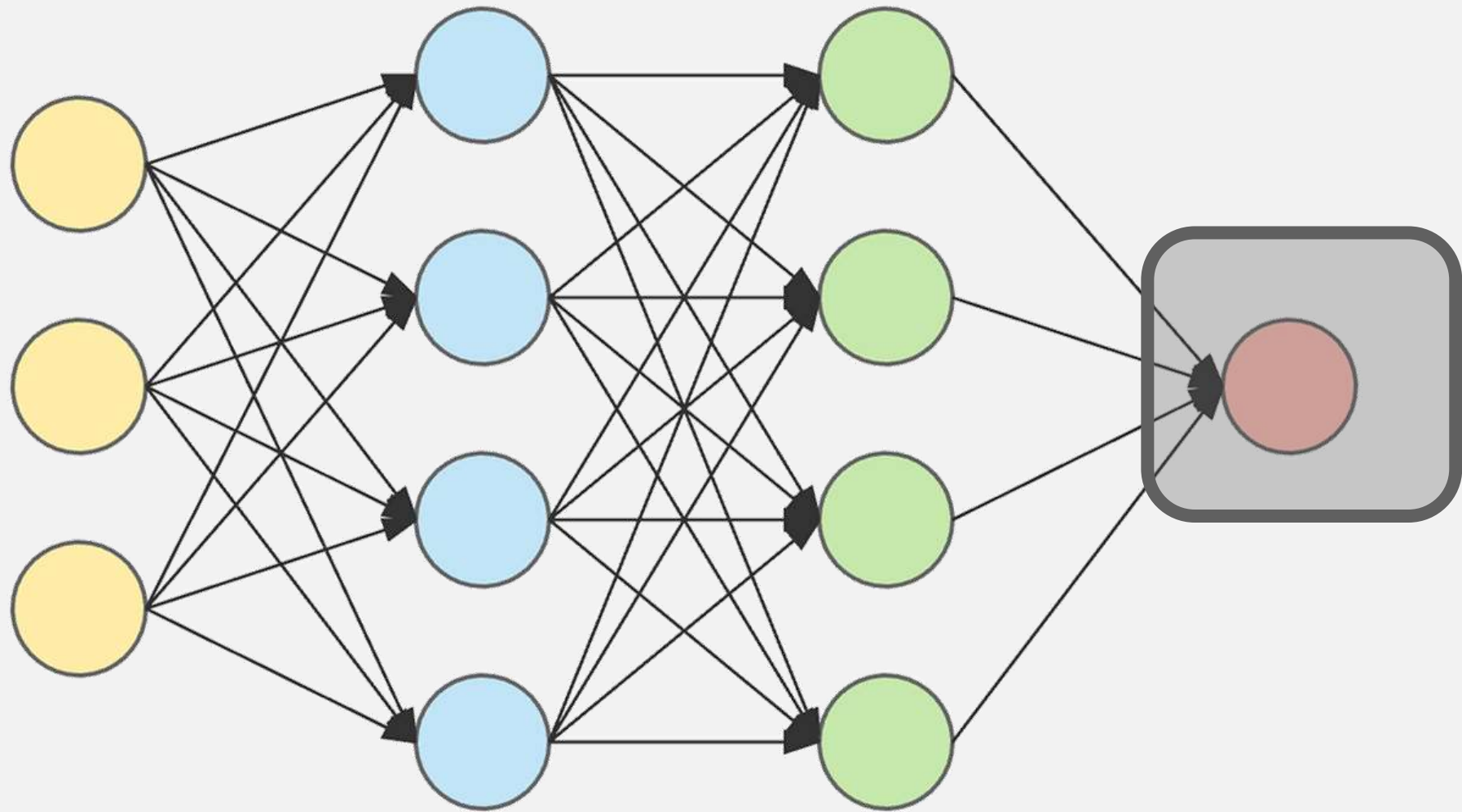
2. Warstwa neuronów / warstwa ukryta / Output 2. : $f(z_i)$

Sieć neuronowa



Wagi: w_i

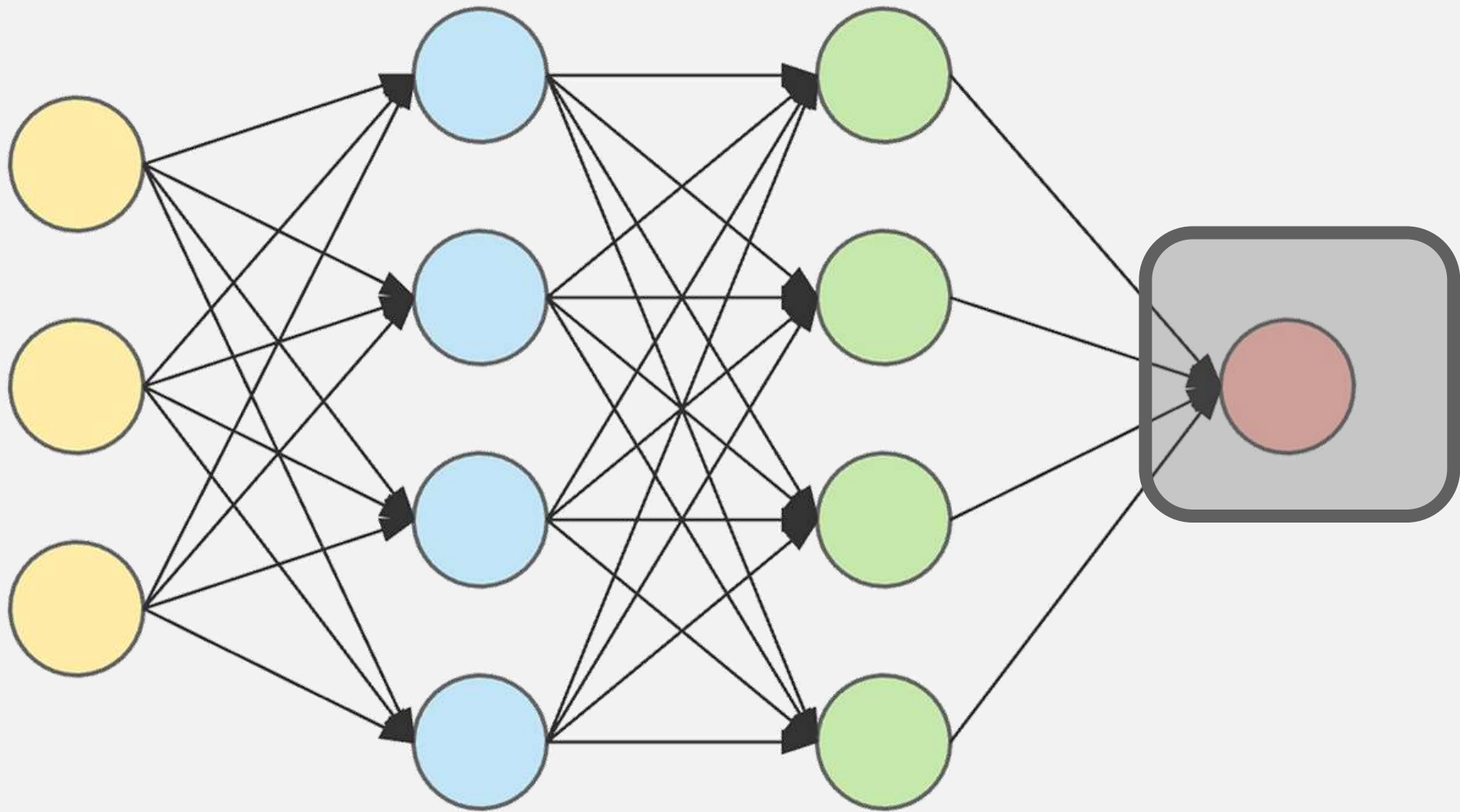
Sieć neuronowa



Warstwa wyjściowa

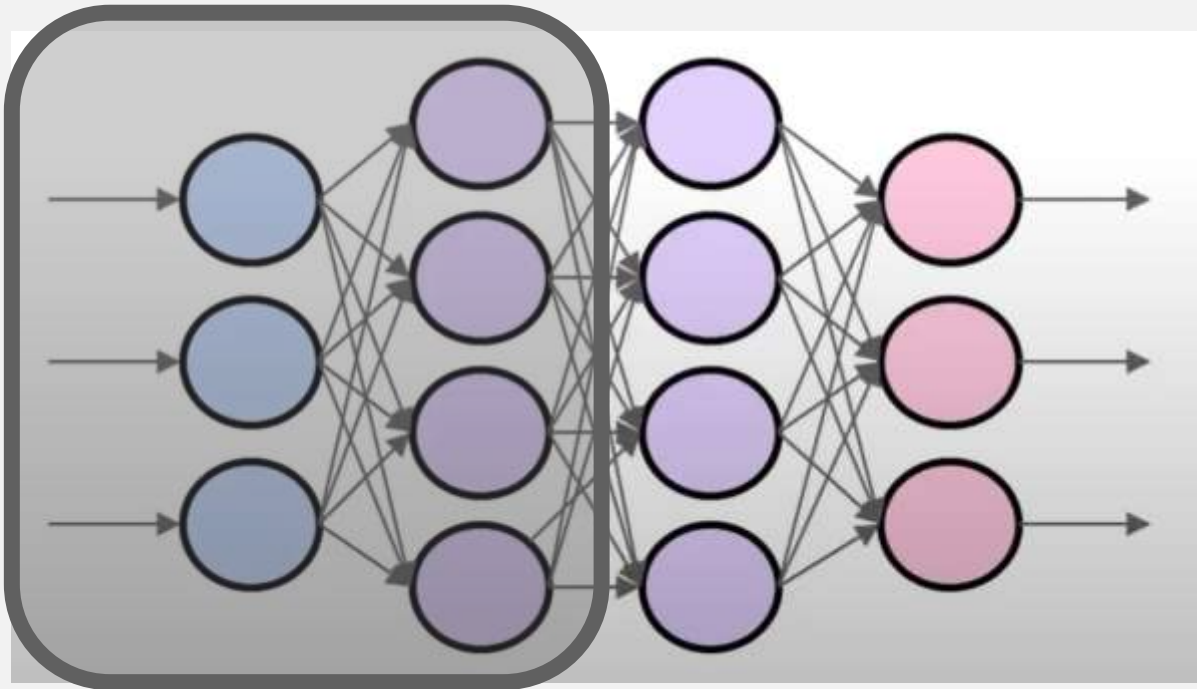
Ostateczny wynik: input dla kolejnej sieci / klasyfikacja / predykcja

Sieć neuronowa



Ostateczny wynik: input dla kolejnej sieci / klasyfikacja / predykcja
DNN / dense neural network / fully connected dense layer

Sieć neuronowa → notacja macierzowa



Dla pojedynczego rekordu danych np. jednego osobnika

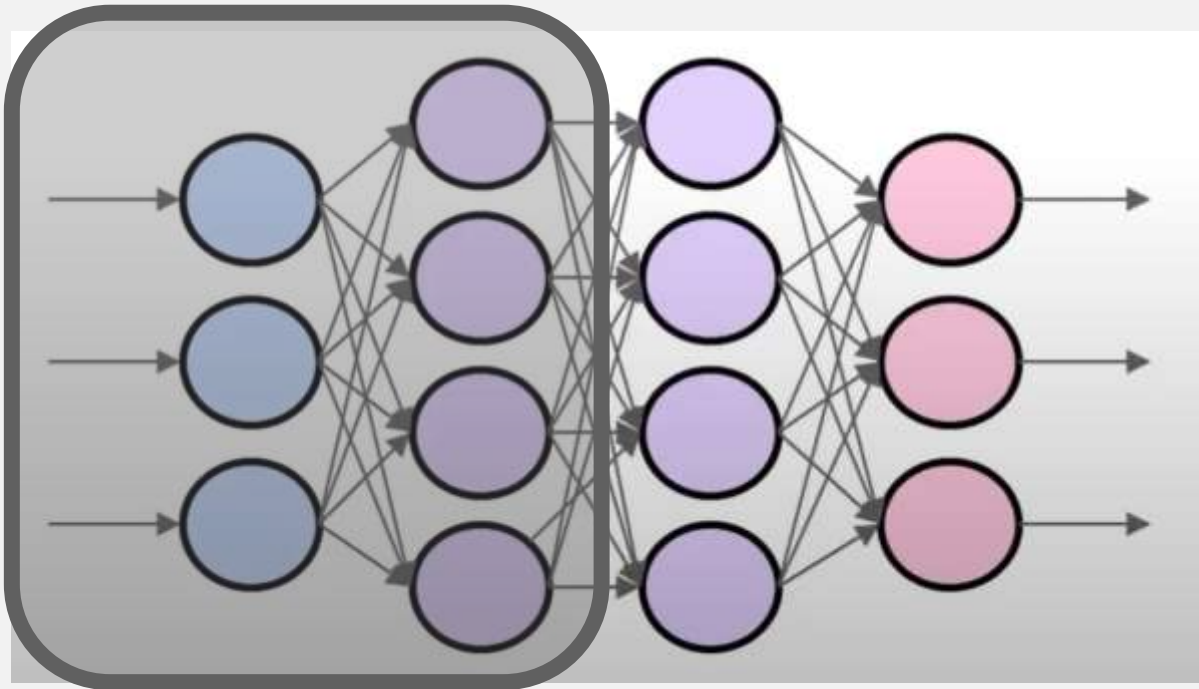
\mathbf{x} [3] wektor wartości wejściowych

\mathbf{W}_1 [3 x 4] macierz wag

\mathbf{z}_1 [4] wektor wyników dla neuronów, $\mathbf{z} = \mathbf{x}\mathbf{W}$

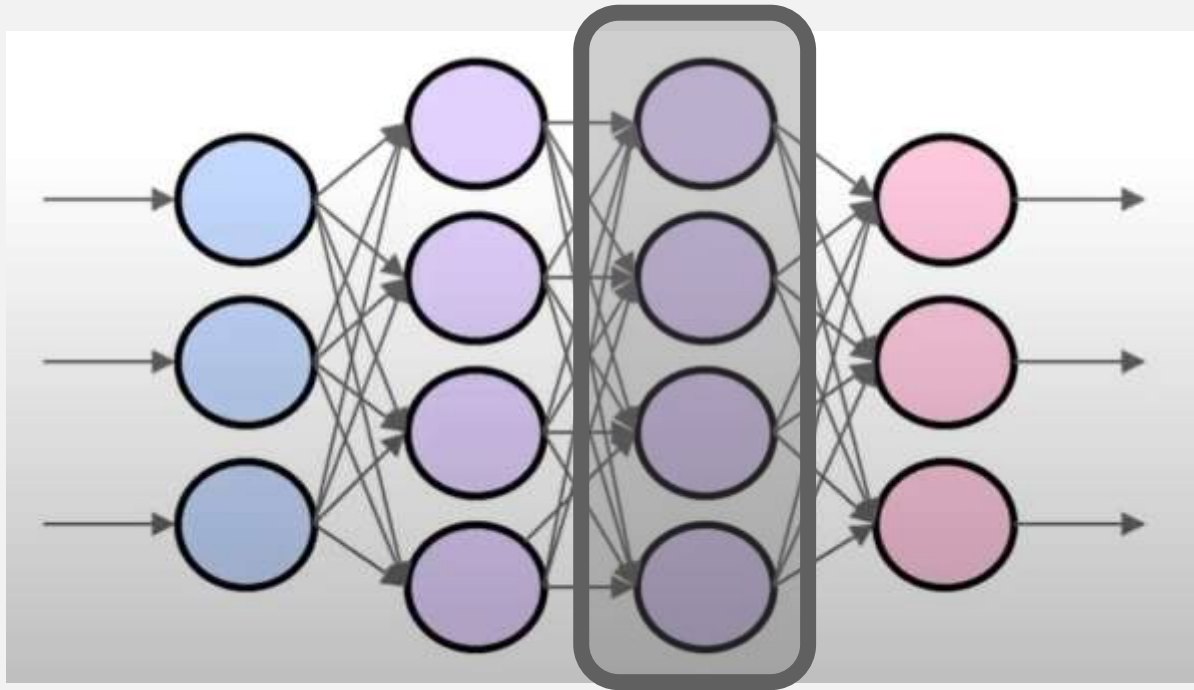
\mathbf{a}_1 [4] wektor wartości funkcji aktywacji, $\mathbf{a} = f(\mathbf{z})$

Sieć neuronowa → notacja macierzowa



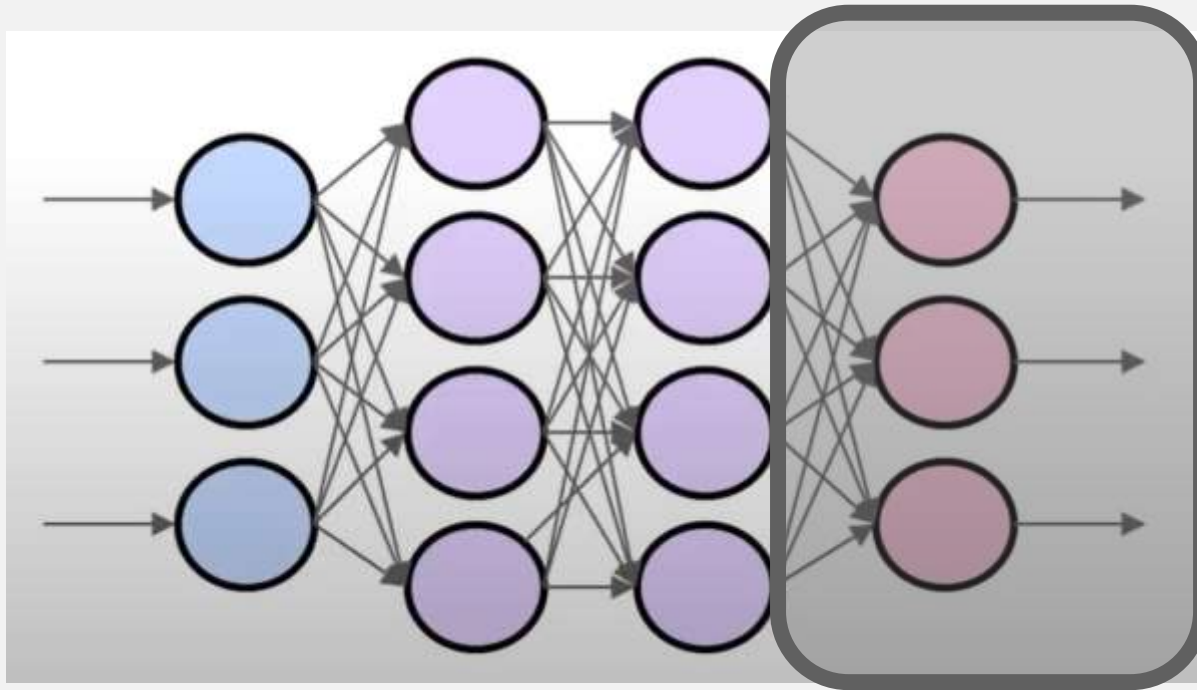
1. $\mathbf{z}_1 = \mathbf{x}W_1 \rightarrow \mathbf{a}_1 = f(\mathbf{z}_1)$
2. $\mathbf{z}_2 = ?$

Sieć neuronowa → notacja macierzowa



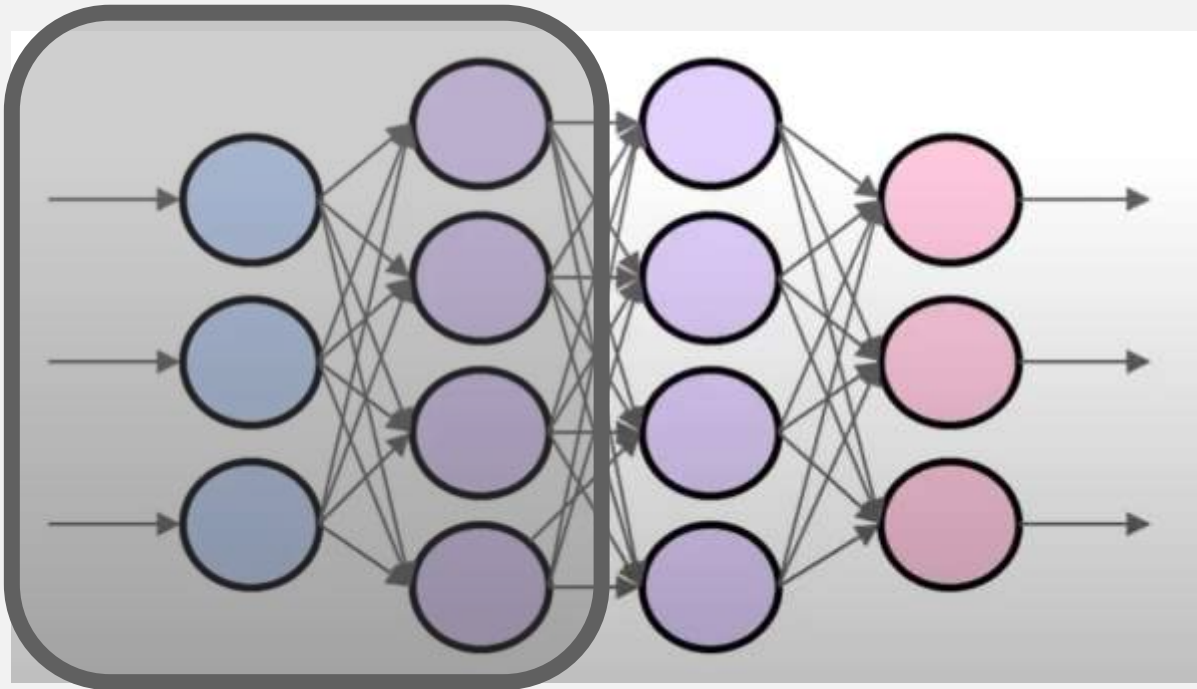
1. $\mathbf{z}_1 = \mathbf{x}\mathbf{W}_1$ $\rightarrow \mathbf{a}_1 = f(\mathbf{z}_1)$
2. $\mathbf{z}_2 = \mathbf{a}_1\mathbf{W}_2$ $\rightarrow \mathbf{a}_2 = f(\mathbf{z}_2)$
3. $\mathbf{z}_3 = ?$

Sieć neuronowa → notacja macierzowa



1. $\mathbf{z}_1 = \mathbf{x}\mathbf{W}_1$ $\rightarrow \mathbf{a}_1 = f(\mathbf{z}_1)$
2. $\mathbf{z}_2 = \mathbf{a}_1\mathbf{W}_2$ $\rightarrow \mathbf{a}_2 = f(\mathbf{z}_2)$
3. $\mathbf{z}_3 = \mathbf{a}_2\mathbf{W}_3$ $\rightarrow \hat{\mathbf{y}} = f(\mathbf{z}_3) = \textit{softmax}(\mathbf{z}_3)$

Sieć neuronowa → notacja macierzowa

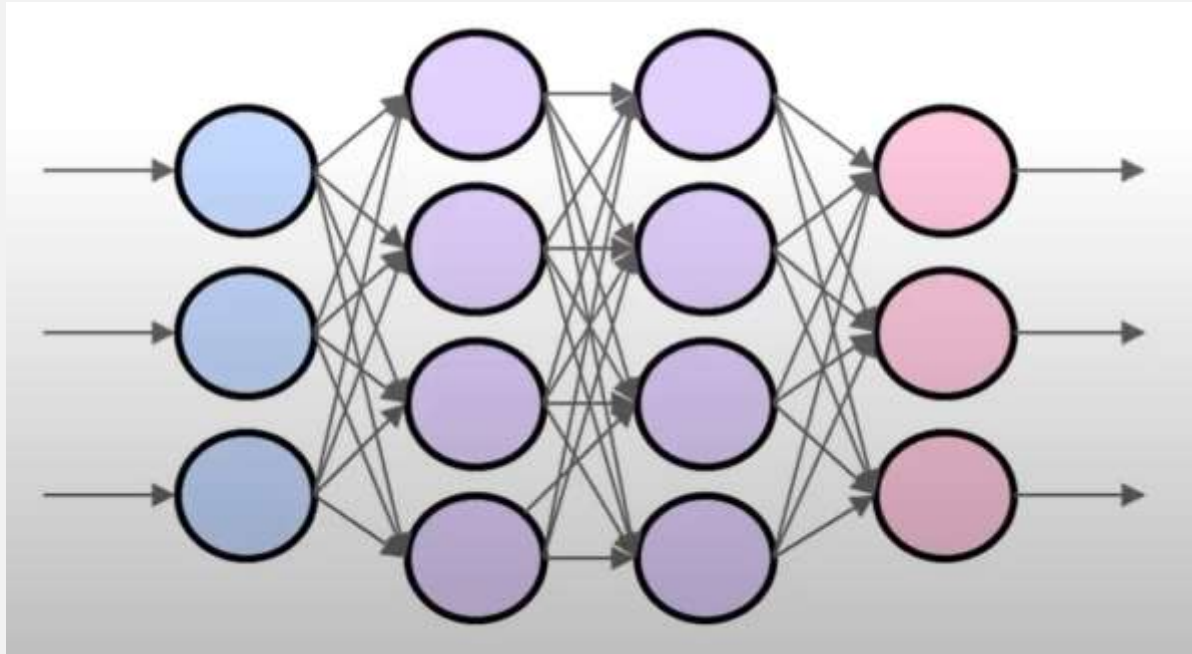


Dla pojedynczego zbioru danych np. wszystkie osobniki

X [N x 3] macierz wartości wejściowych

\hat{Y} [N x 3] macierz wartości wyjściowych

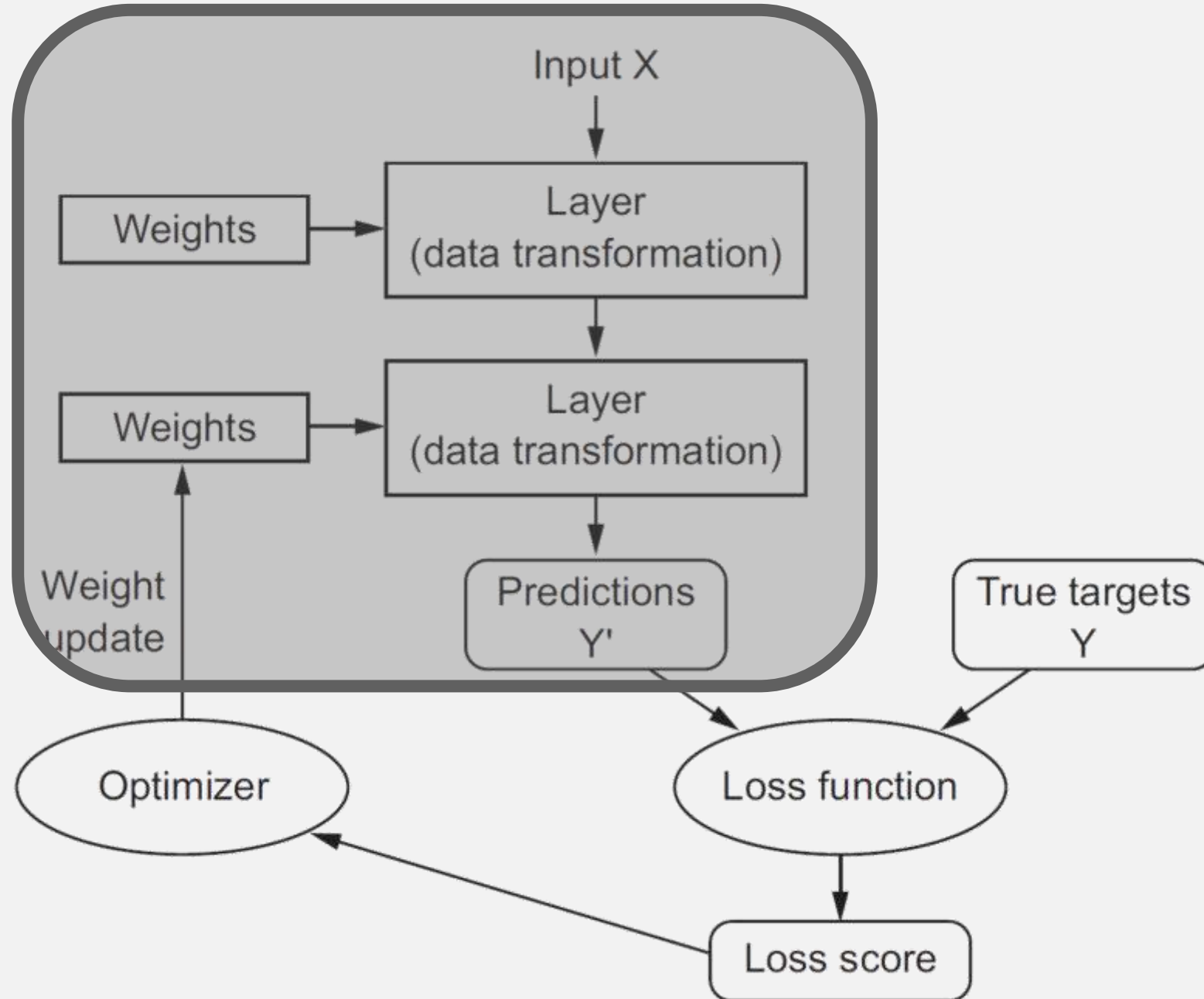
Sieć neuronowa → implementacja keras



```
from keras import models
from keras import layers
```

```
model = models.Sequential()
model.add(layers.Dense(4, activation='relu', input_shape=(N,)))
model.add(layers.Dense(4, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))
```


Architektura modelu uczenia głębokiego



1. Pojedynczy neuron
2. Przykładowe funkcje aktywacji
3. Pojedyncza sieć neuronowa
4. Zestaw sieci neuronowych
5. Architektura modelu uczenia głębokiego

1. Co może stanowić dane wejściowe dla pojedynczego neuronu?
2. W którym miejscu neuronu stosuje się funkcję sigmoid?
3. Jakie komponenty zawiera model uczenia głębokiego?