

## **Użycie SVD do rozkładów obrazów**

### **Zadanie Twarze Własne**

Sprawozdanie z ćwiczeń  
Matematyka Konkretna

Data wykonania:  
28.06.2025

Autor:  
Bartosz Bieniek 058085

## 1. Cel ćwiczenia

Funkcje aktywacji w sieciach neuronowych, Poprowadzić badania funkcji aktywacji zgodnie z wariantem zadania z użyciem Python. Obliczyć gradient funkcji. Wyświetlić funkcję wraz z gradientem na jednym wykresie. We wniosku opisać zagadnienia w których używają daną funkcję aktywacji. Zadanie umieścić na [Github](#).

## 2. Przebieg ćwiczenia

### 1. Import bibliotek

Zaimportowano biblioteki numpy oraz matplotlib.pyplot, niezbędne do przeprowadzenia analizy numerycznej i graficznej. numpy umożliwiło wektorowe obliczenia wartości funkcji oraz jej pochodnej, a matplotlib wykorzystano do wizualizacji przebiegu obu funkcji. Dzięki temu przygotowano środowisko do badania właściwości funkcji aktywacji.

```
import numpy as np
import matplotlib.pyplot as plt
```

[1] ✓ 0.2s

Rys. 1. Import bibliotek

### 2. Definicja funkcji sigmoidalnej oraz jej pochodnej

Zdefiniowano funkcję sigmoidalną jako odwrotność sumy:  $1+e^{-x}$ , co jest standardową postacią tej funkcji. Dodatkowo wyprowadzono jej analityczną pochodną, korzystając z tożsamości:  $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$ . Obie funkcje zapisano w postaci programistycznej, co umożliwiło ich dalszą analizę i wizualizację.

```
# Funkcja sigmoidalna
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Pochodna funkcji sigmoidalnej
def sigmoid_derivative(x):
    s = sigmoid(x)
    return s * (1 - s)
```

[2] ✓ 0.0s Python

Rys. 2. Definicja funkcji sigmoidalnej oraz jej pochodnej

### 3. Zakres danych wejściowych do analizy

Zdefiniowano wektor wartości wejściowych  $x$  w przedziale od -10 do 10, co umożliwiło obserwację pełnego przebiegu funkcji i jej pochodnej. Obliczono wartości funkcji sigmoidalnej oraz jej pochodnej dla każdego punktu w tym zakresie. Uzyskane dane posłużyły do wygenerowania wykresów porównawczych.

```
x = np.linspace(-10, 10, 500)
y = sigmoid(x)
dy = sigmoid_derivative(x)
```

Rys. 3. Zakres danych wejściowych do analizy

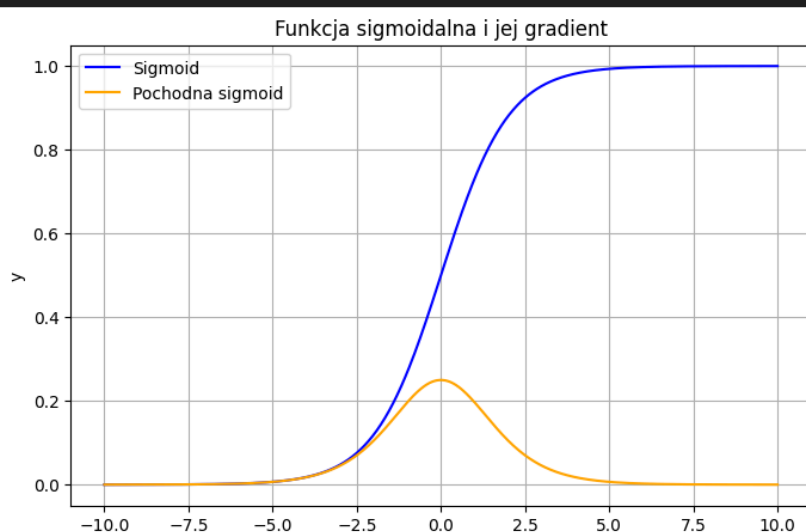
### 4. Wykres funkcji i jej gradientu

Na jednym wykresie przedstawiono przebieg funkcji sigmoidalnej oraz jej pochodnej, co umożliwiło bezpośrednie porównanie ich właściwości. Funkcja aktywacji osiąga asymptoty przy wartościach 0 i 1, natomiast jej pochodna ma maksimum w punkcie  $x = 0$ . Wykres zilustrował typowe cechy funkcji sigmoid – nieliniowość oraz ograniczenie gradientu dla dużych wartości bezwzględnych wejścia.

```
plt.figure(figsize=(8, 5))
plt.plot(x, y, label='Sigmoid', color='blue')
plt.plot(x, dy, label='Pochodna sigmoid', color='orange')
plt.title("Funkcja sigmoidalna i jej gradient")
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True)
plt.legend()
plt.show()
```

[4] ✓ 0.1s

Python



Rys. 4. Wykres funkcji i jej gradientu

### **3. Wnioski**

Funkcja sigmoidalna jest nieliniową funkcją aktywacji, która przekształca dowolne wejście do zakresu  $(0, 1)$ , co czyni ją przydatną w zadaniach klasyfikacji binarnej, szczególnie w warstwach wyjściowych modeli.

Jej pochodna osiąga maksimum w pobliżu zera, a dla dużych wartości wejściowych zbliża się do zera, co skutkuje zjawiskiem zanikającego gradientu i ogranicza jej zastosowanie w głębokich sieciach.