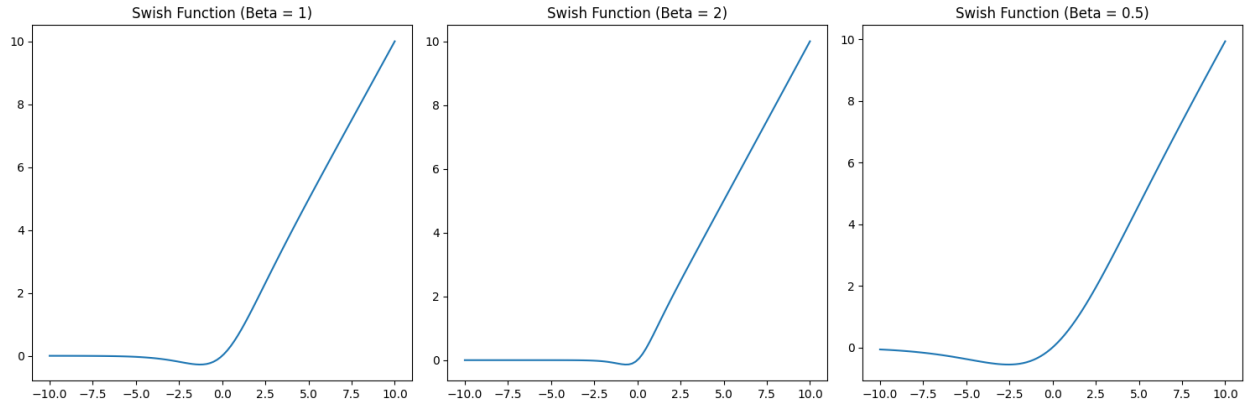
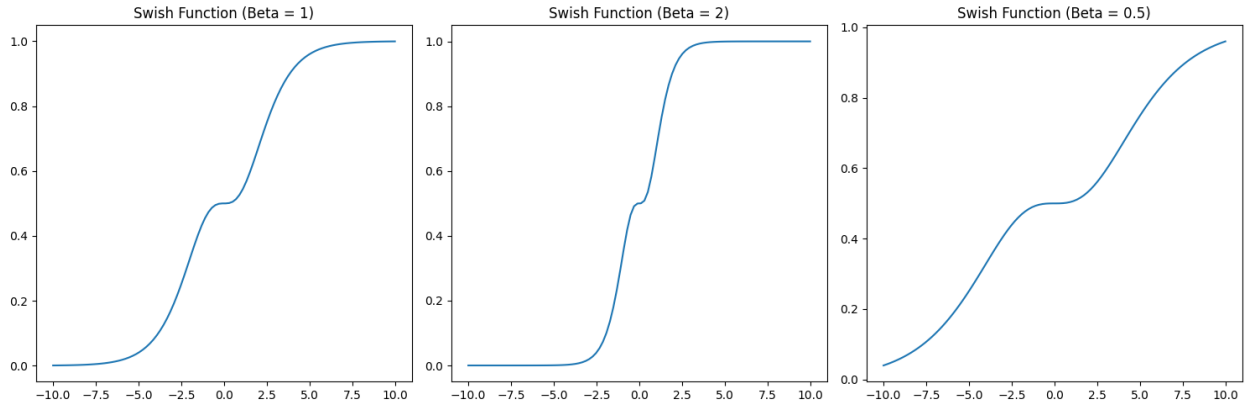
```python
# Lab 6, Rafał Klinowski
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def swish(x, beta):
    return x * sigmoid(beta * x)

# Wykres funkcji
import matplotlib.pyplot as plt

xs = np.linspace(-10,10,100)
y1 = swish(xs, 1)
y2 = swish(xs, 2)
y3 = swish(xs, 0.5)

# Trzy wykresy
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(xs, y1)
plt.title('Swish Function (Beta = 1)')

plt.subplot(1, 3, 2)
plt.plot(xs, y2)
plt.title('Swish Function (Beta = 2)')

plt.subplot(1, 3, 3)
plt.plot(xs, y3)
plt.title('Swish Function (Beta = 0.5)')

plt.tight_layout()
plt.show()
```




```python
# Gradient funkcji
# d/dx (swish(x, beta)) = x' * (1 + e^-Bx) + x * (1 + e^-Bx)' / (1 +
```

```
e^-Bx)^2
def swish_gradient(x, beta):
    return ((1 + np.exp(-beta * x)) + x * (-beta * np.exp(-beta * x)))
/ (1 + np.exp(-beta * x))**2

xs = np.linspace(-10,10,100)
ygradient1 = swish_gradient(xs, 1)
ygradient2 = swish_gradient(xs, 2)
ygradient3 = swish_gradient(xs, 0.5)

# Trzy wykresy
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.plot(xs, ygradient1)
plt.title('Swish Function (Beta = 1)')

plt.subplot(1, 3, 2)
plt.plot(xs, ygradient2)
plt.title('Swish Function (Beta = 2)')

plt.subplot(1, 3, 3)
plt.plot(xs, ygradient3)
plt.title('Swish Function (Beta = 0.5)')

plt.tight_layout()
plt.show()
```




```
# Wyświetlenie funkcji z gradientem na jednym wykresie
plt.figure(figsize=(15, 5))

xs = np.linspace(-5,5,100)
y1 = swish(xs, 1)
ygradient1 = swish_gradient(xs, 1)

plt.plot(xs, y1, 'b', label='Swish Function')
plt.plot(xs, ygradient1, 'r', label='Gradient')
```

```
plt.title('Swish Function (Beta = 1)')
plt.legend(loc='upper left')
plt.show()
```