Sprawozdanie

Bartosz Jabłoński – PRiR, laboratoria 10

**Zadanie 1**

def prostokatow(function, a, b, i):

    dx = (b - a) / i

    integr = 0

    for x in range(i):

        x = x \* dx + a

        integr += dx \* eval(function)

    return integr

integral = simpson(lambda x: x\*\*2, 0.0, 1.0, 100)

def simpson(my\_func, a, b, n):

  delta\_x = (b-a)/n

  total = 0

  for i in range(0, n, 2):

    x = a + delta\_x \* 2 \* i

    total += delta\_x \* (my\_func(x) + 4 \* my\_func(x + delta\_x) + my\_func(x + 2 \* delta\_x)) / 3

  return total

integral = simpson(lambda x: x\*\*2, 0.0, 1.0, 100)

def trapezow(function, a, b, i):

    dx = (b - a) / i

    integr = 0

    for x in range(i):

        x = x \* dx + a

        fx1 = eval(function)

        x += dx

        fx2 = eval(function)

        integr += 0.5 \* dx \* (fx1 + fx2)

    return integr

integral = trapezow('x\*\*2', 0.0, 1.0, 100)

**Zadanie 2**

import tensorflow as tf

import numpy as np

from tensorflow import keras

model = tf.keras.Sequential([keras.layers.Dense(units=1, input\_shape=[1])])

model.compile(optimizer='sgd', loss='mean\_squared\_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)

ys = np.array([0.0, 8.0, 6.0, -0.0, -4.0, 0.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([0.0]))

**Zadanie 3**

num\_rows = 5

num\_cols = 3

num\_images = num\_rows\*num\_cols

plt.figure(figsize=(2\*2\*num\_cols, 2\*num\_rows))

for i in range(num\_images):

   plt.subplot(num\_rows, 2\*num\_cols, 2\*i+1)

   plot\_image(i, predictions[i], test\_labels, test\_images)

   plt.subplot(num\_rows, 2\*num\_cols, 2\*i+2)

   plot\_value\_array(i, predictions[i], test\_labels)

plt.tight\_layout()

plt.show()

