ZAD 1

```
import tensorflow as tf
```

```
x = tf.Variable(4.0)
y = tf.Variable(3.0)

with tf.GradientTape() as tape:
    f = (x**3)+(y**2)
    df_dx,df_dy = tape.gradient(f,(x,y))

print(df_dx.numpy())
print(df_dy.numpy())
```

```
48.0
6.0
```

ZAD 2

```
x = tf.Variable(1.0)
y = tf.Variable(2.0)

with tf.GradientTape() as tape:
    f = 4*(x**3)+11*(y**2)+9*y*x+10
    df_dx,df_dy = tape.gradient(f,(x,y))

print(df_dx.numpy())
print(df_dy.numpy())
```

Zad 3

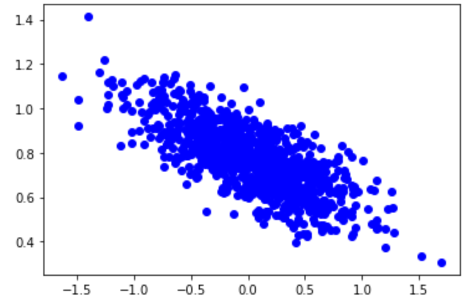
```
import matplotlib.pyplot as plt
import numpy as np
```

```
number_of_points = 1000
x_point = []
y_point = []
```

```
a = -0.22
b = 0.78
```

```
for i in range(number_of_points):
    x = np.random.normal(0.0,0.5)
    y = (a*x+b)+np.random.normal(0.0,0.1)
    x_point.append(x)
    y_point.append(y)
```

```
plt.scatter(x_point,y_point,c='b')
plt.show()
```



```
real_x = np.array(x_point)
real_y = np.array(y_point)
```

```
def loss_fn(real_y, pred_y):
    return tf.reduce_mean((real_y - pred_y)**2)
```

```
import random
a·=·tf.Variable(random.random())
b·=·tf.Variable(random.random())
```

```
Loss = []
epochs = 1000
learning_rate = 0.01

for _ in range(epochs):
  with tf.GradientTape() as tape:
    pred_y = a * real_x + b
    loss = loss_fn(real_y, pred_y)
    Loss.append(loss.numpy())

  dloss_da, dloss_db = tape.gradient(loss,(a, b))

  a.assign_sub(learning_rate*dloss_da)  #a = a - alpha*dloss_da
  b.assign_sub(learning_rate*dloss_db)  #b = b - alpha*dloss_db
```

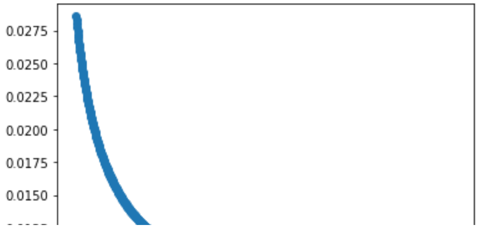
```
np.max(Loss),np.min(Loss)
```

```
(0.028574357, 0.0098474445)
```

```
print(a.numpy())
print(b.numpy())
```

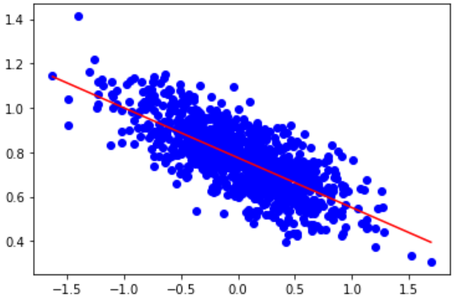
```
-0.22509728
0.77586883
```

```
plt.scatter(np.arange(epochs),Loss)
plt.show()
```



```
max = np.max(x_point)
min = np.min(x_point)


X = np.linspace(min, max, num=10)
plt.plot(X,a.numpy()*X+b.numpy(),c='r')
plt.scatter(x_point,y_point,c="b")
plt.show()
```



```
def subset_dataset(x_dataset, y_dataset, subset_size):
    arr = np.arange(len(x_dataset))
    np.random.shuffle(arr)
    x_train = x_dataset[arr[0:subset_size]]
    y_train = y_dataset[arr[0:subset_size]]
    return x_train,y_train


a = tf.Variable(random.random())
b = tf.Variable(random.random())


Loss = []
epochs = 1000
learning_rate = 0.2
batch_size = 50

for i in range(epochs):
  real_x_batch,real_y_batch = subset_dataset(real_x,real_y,batch_size)
  with tf.GradientTape() as tape:
    pred_y = a * real_x_batch + b
    loss = loss_fn(real_y_batch, pred_y)
    Loss.append(loss.numpy())

  dloss_da, dloss_db = tape.gradient(loss,(a, b))
```

```
    a.assign_sub(learning_rate*dloss_da)  #a = a - alpha*dloss_da
    b.assign_sub(learning_rate*dloss_db)  #b = b - alpha*dloss_db
```

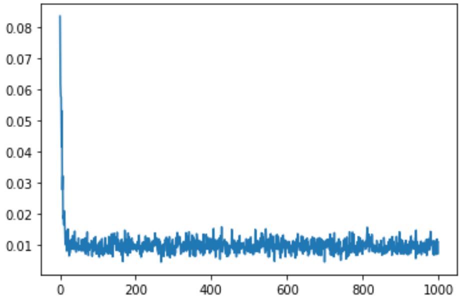
```
np.max(Loss),np.min(Loss)
```

```
(0.0836345, 0.0046261256)
```

```
print(a.numpy())
print(b.numpy())
```

```
-0.21703197
0.77252394
```

```
plt.plot(Loss)
plt.show()
```



✓ 0 s ukończono o 11:35