Zadanie 1

```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import random
```

```
import keras
from keras.models import Sequential
from keras.layers import Dense
```

Two gangs

Dataset:

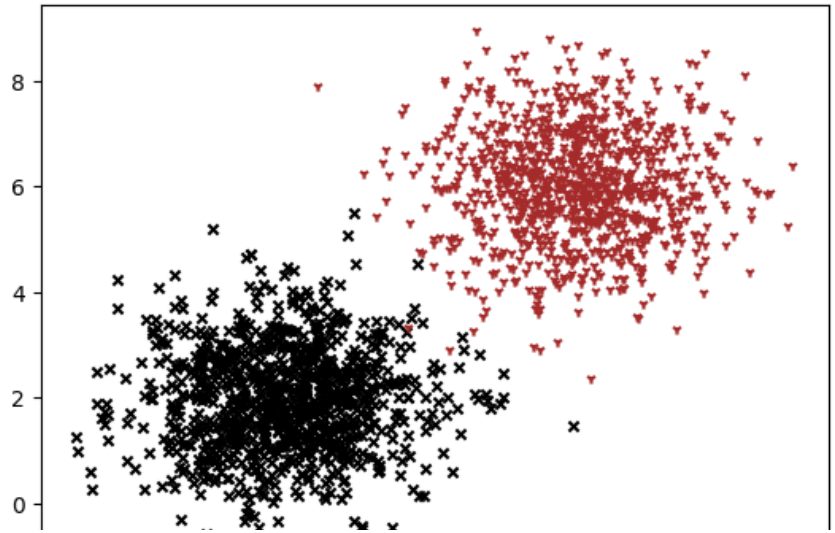
```
[0]*10+[1]*10
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
x_label1 = np.random.normal(3, 1, 1000)
y_label1 = np.random.normal(2, 1, 1000)
x_label2 = np.random.normal(7, 1, 1000)
y_label2 = np.random.normal(6, 1, 1000)

xs = np.append(x_label1, x_label2)
ys = np.append(y_label1, y_label2)
labels = np.asarray([0.]*len(x_label1)+[1.]*len(x_label2))
labels
```

```
array([0., 0., 0., ..., 1., 1., 1.])
```

```
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='brown', marker='1', s=20)
plt.show()
```




```
x_label1
```

```
              3.55988207, 2.3028118 , 2.44941217, 2.66893442, 1.16583699,
              3.76068076, 2.99390727, 2.16712186, 2.31190459, 2.33751105,
              3.67840779, 1.42488343, 2.54661594, 3.92247047, 2.69736482,
              3.39710663, 3.05190111, 2.65005557, 3.75353397, 2.87846765,
              1.50801301, 3.13558611, 1.35969293, 4.37660503, 2.21013076,
              1.81105005, 2.51165079, 2.58229562, 2.21173423, 1.86441822,
              3.10225181, 1.90786467, 2.09869339, 3.52734798, 1.26052506,
              1.99705453, 3.80791401, 4.53185964, 1.80592313, 2.51307418,
              4.00505919, 3.26586071, 2.79768268, 3.18836062, 1.94577422,
              2.45916503, 3.53118182, 1.55233525, 3.33730109, 3.42256693,
              3.42014593, 2.72861821, 3.98575827, 4.10150746, 2.17669025,
              3.85672411, 2.75784226, 2.12889833, 3.20512885, 2.7663865 ,
              2.39651873, 3.63521304, 2.88890127, 1.9790647 , 1.60355282,
              0.46879331, 2.28078477, 3.54685527, 2.42120533, 2.45168469,
              4.12054673, 3.81607588, 4.82517406, 2.58336502, 0.98000369,
              3.51514254, 2.85819216, 2.24821881, 4.06941353, 3.13234499,
              2.85479912, 4.76729806, 3.15452528, 3.0160224 , 2.88278267,
              2.83421687, 4.66737627, 2.06793176, 3.48717193, 4.19919275,
              5.05657023, 3.5579438 , 1.75169028, 3.16095045, 3.1769958 ,
              2.3831249 , 2.92489028, 3.79220009, 3.99641716, 2.60076869,
              1.77370885, 3.48719661, 2.57274896, 4.21547842, 2.58920809,
              1.53543394, 3.41725187, 3.07964999, 2.76962407, 3.30547422,
              4.54398408, 4.43963239, 1.20062937, 2.27437446, 3.33287217,
              3.48271719, 3.71079505, 1.52941407, 3.13373179, 3.56355977,
              4.26614829, 4.20487664, 3.67698166, 2.94760193, 3.96018349,
              2.60853081, 1.91507396, 3.41018161, 5.01799022, 3.64985342,
              2.74505619, 4.5484449 , 3.66426771, 4.36329257, 3.4228162 ,
              1.1058157 , 3.32314545, 2.77719083, 2.38270548, 1.79099726,
              2.90126257, 2.04523166, 3.20782643, 1.73617873, 3.62778675])
```

```
def loss_fn_grad(y, y_model):
 return tf.reduce_mean(-y*tf.math.log(y_model)-(1-y)*tf.math.log(1-y_model))
```

```
def subset_dataset(x_dataset, y_dataset,label,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]]
    label_train = label[arr[0:subset_size]]
    return x_train,y_train,label_train
```

```
labels.shape
```

```
(2000,)
```

```
Loss = []
epochs = 1000
learning_rate = 0.01
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

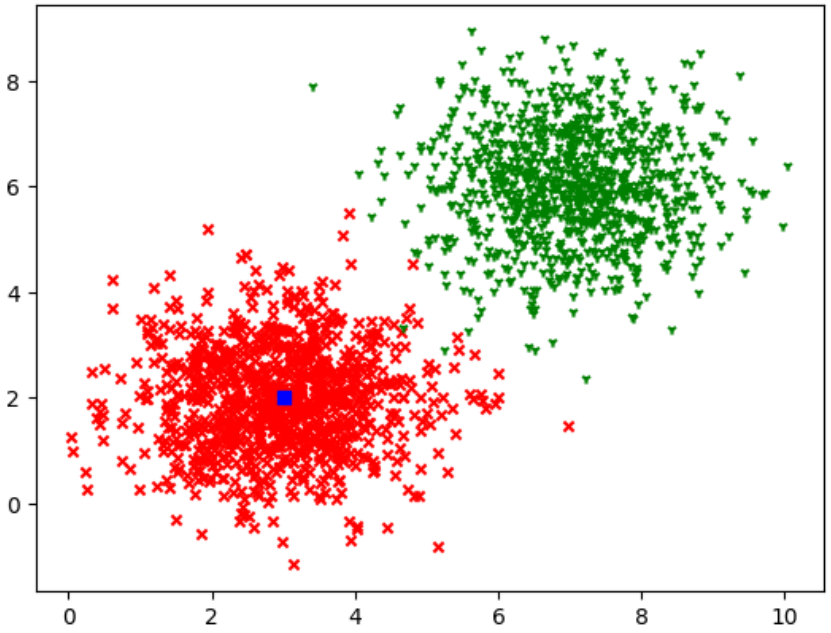
```
(1.521387, 0.30620345)
```

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

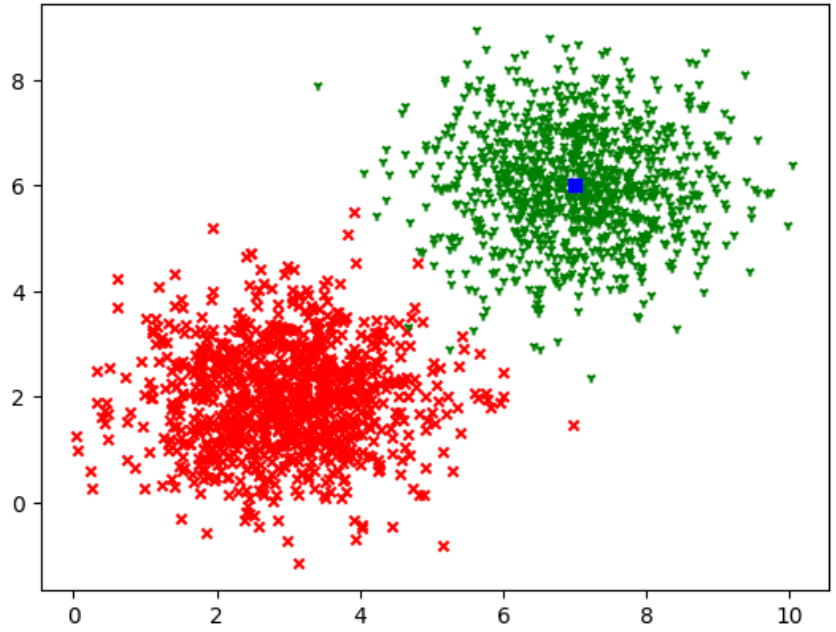
```
-0.03411645
0.53702235
```

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

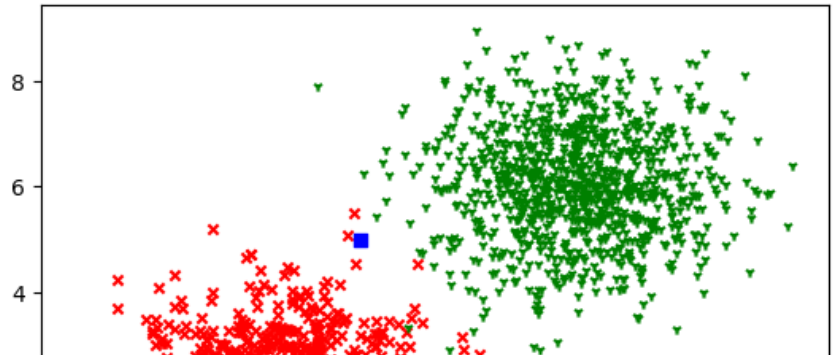

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Hiperparametria

Learning rate 0.1

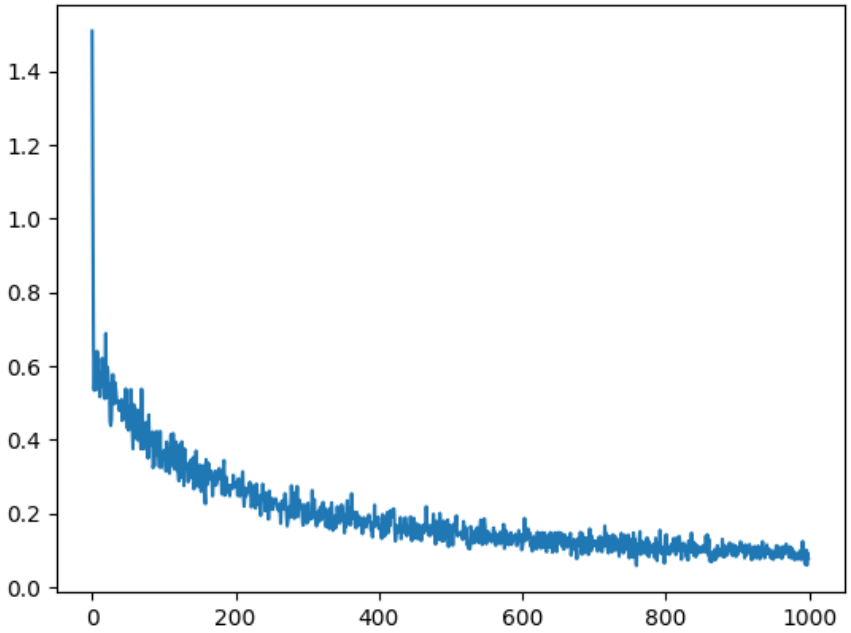
```
Loss = []
epochs = 1000
learning_rate = 0.1
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(1.5082442, 0.058400158)
```

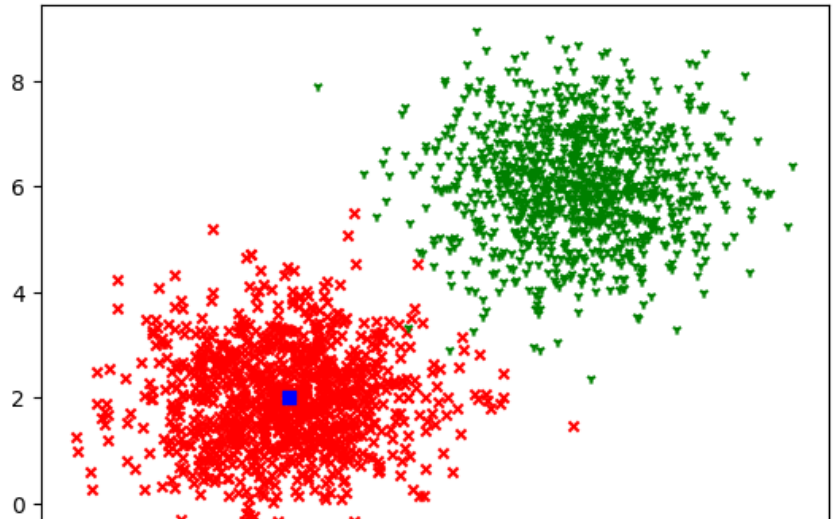
```
print(a.numpy())
print(b.numpy())
```

```
0.5717125
0.87403923
```

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

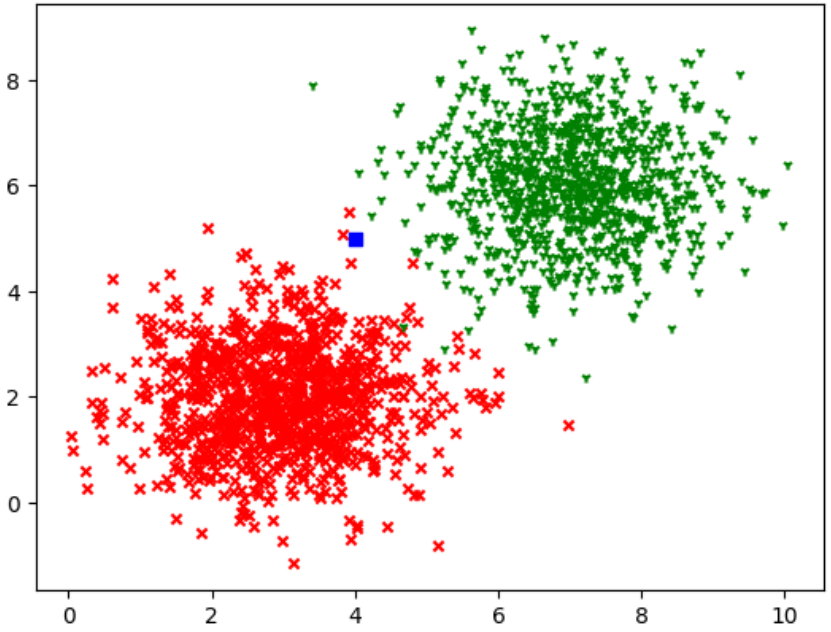



```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```

```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Learning rate 0.001

```
Loss = []
epochs = 1000
learning_rate = 0.001
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

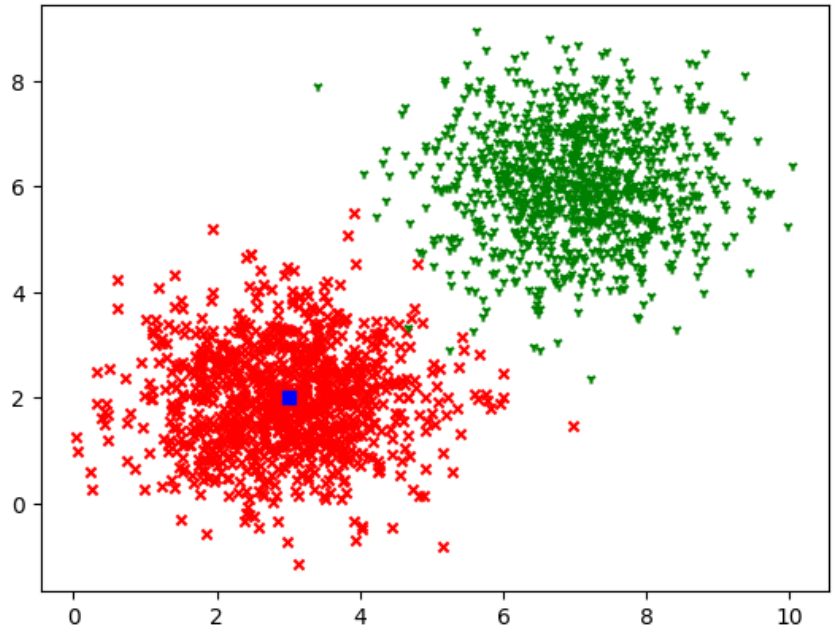
```
(3.0714028, 0.44047368)
```

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

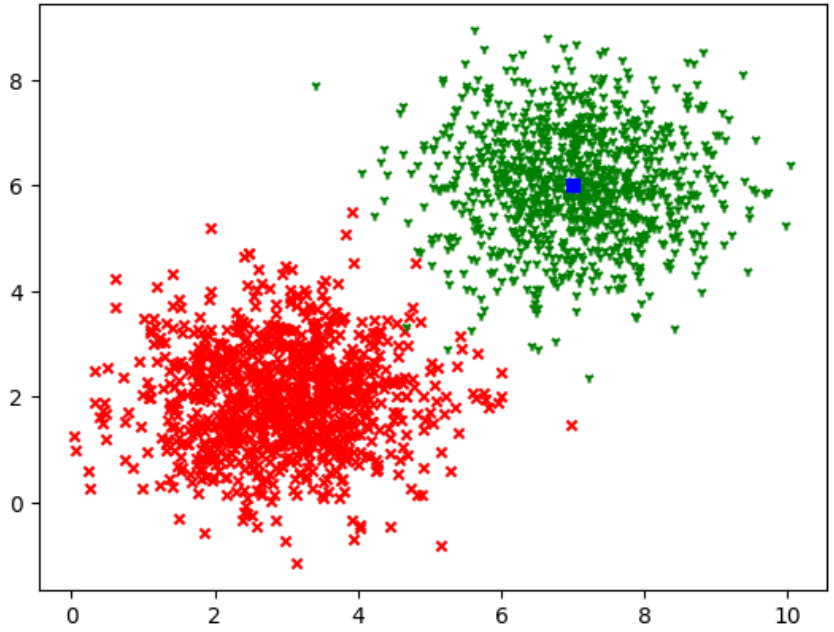
```
-0.03317762
0.2501896
```

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

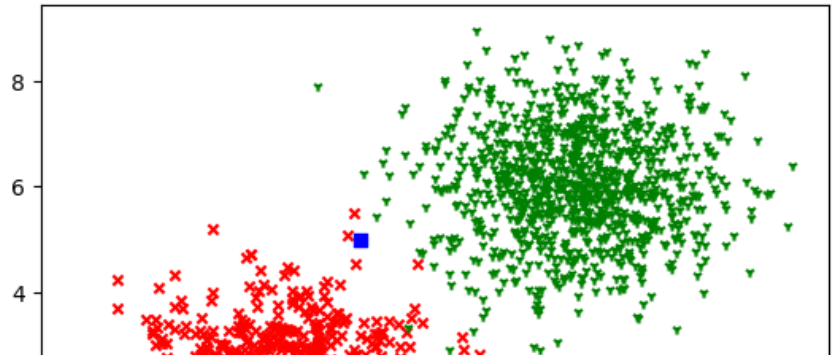

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Number of epchos - 100

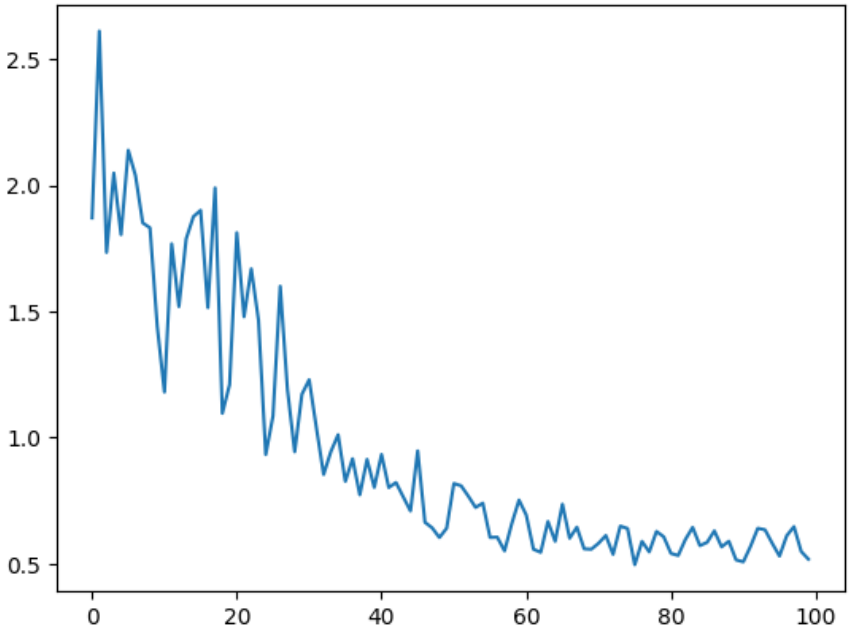
```
Loss = []
epochs = 100
learning_rate = 0.01
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(2.610952, 0.49552184)
```

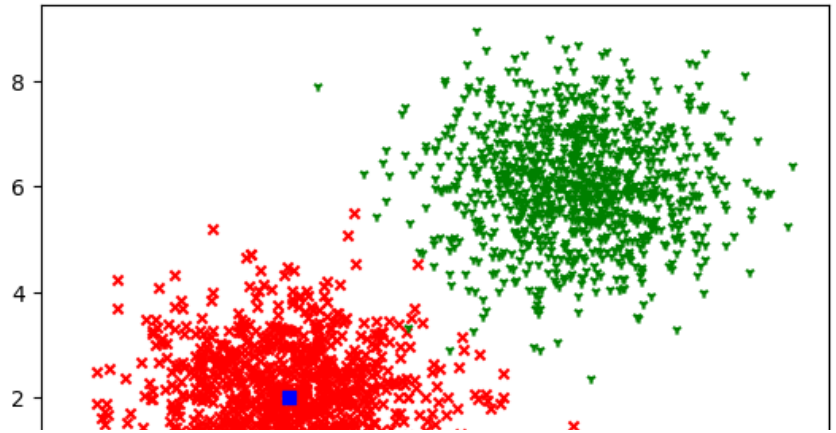
```
print(a.numpy())
print(b.numpy())
```

```
0.097366445
0.11506871
```

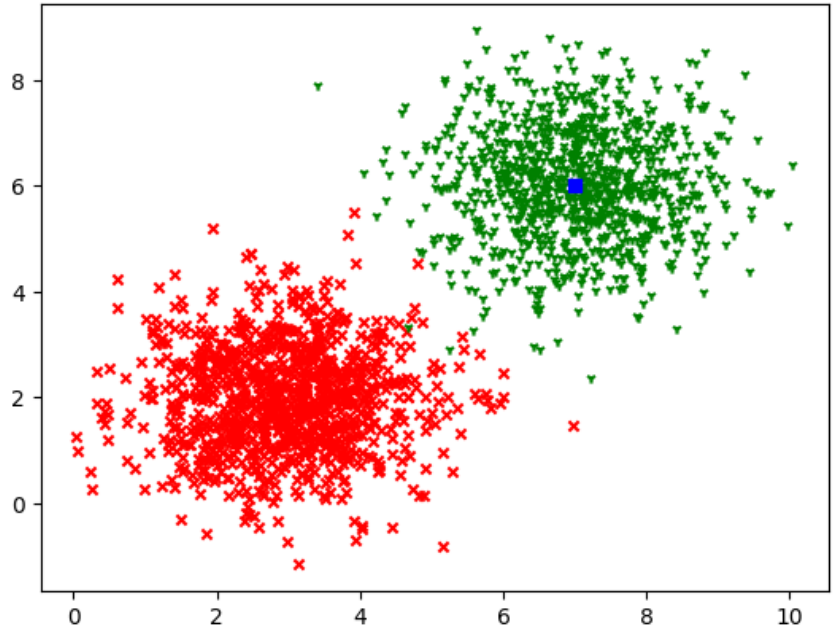
```
plt.plot(Loss)
plt.show()
```



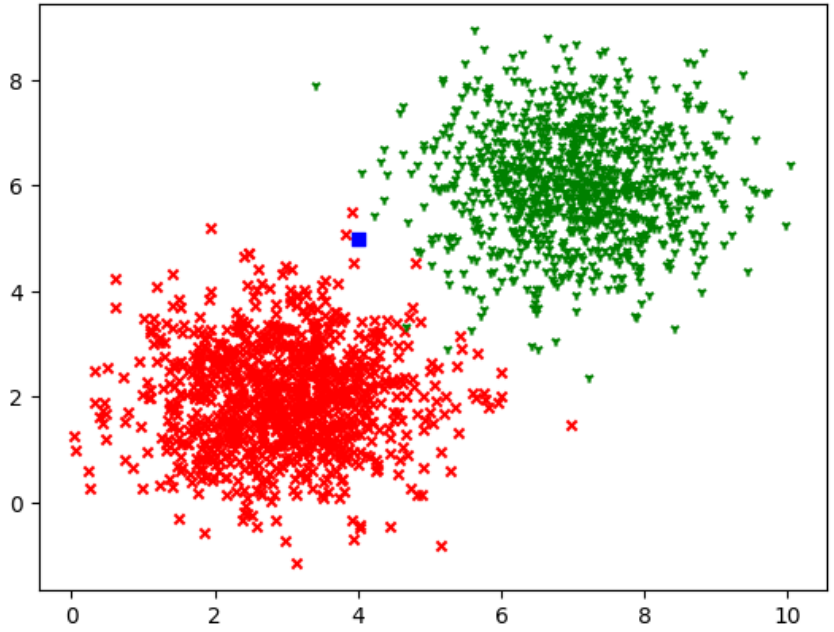

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Number of epochs - 3000

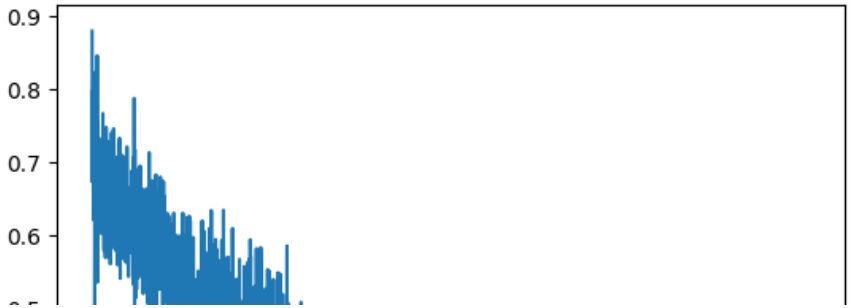
```
Loss = []
epochs = 3000
learning_rate = 0.01
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(0.8797482, 0.14934888)
```

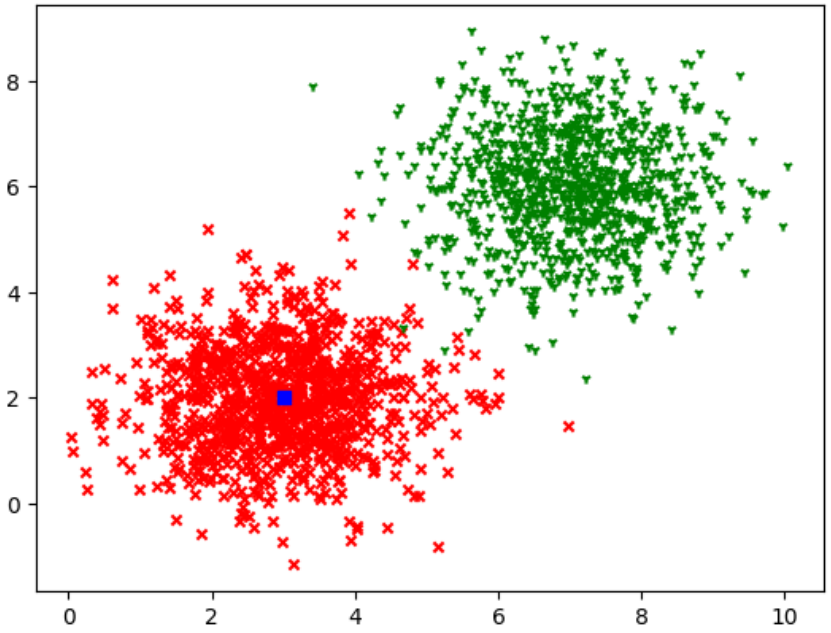
```
print(a.numpy())
print(b.numpy())
```

```
0.15790226
0.6693555
```

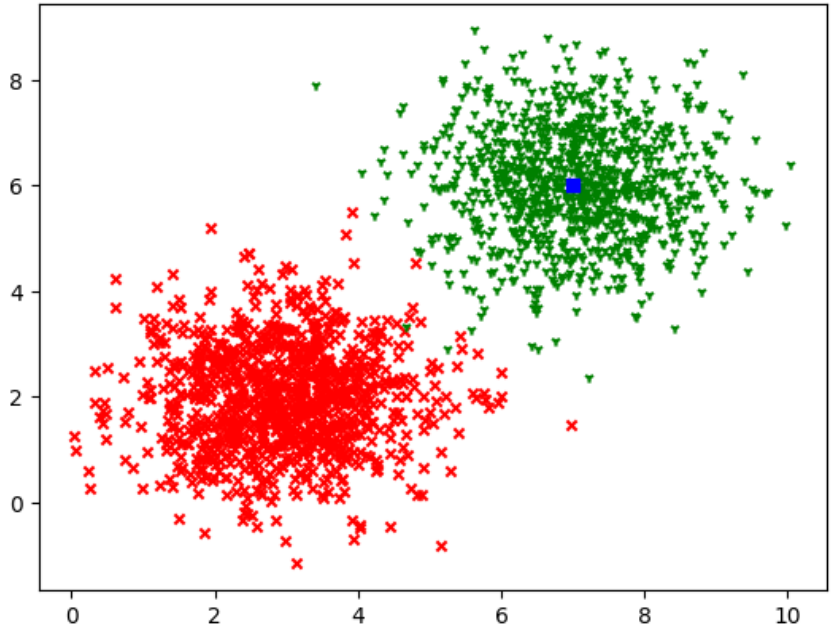
```
plt.plot(Loss)
plt.show()
```



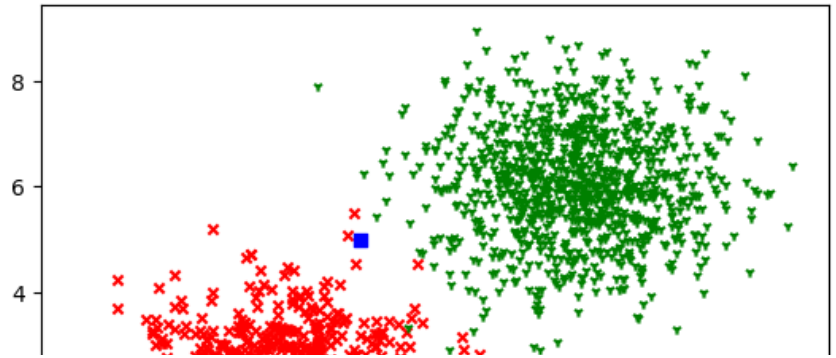

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Minibatch

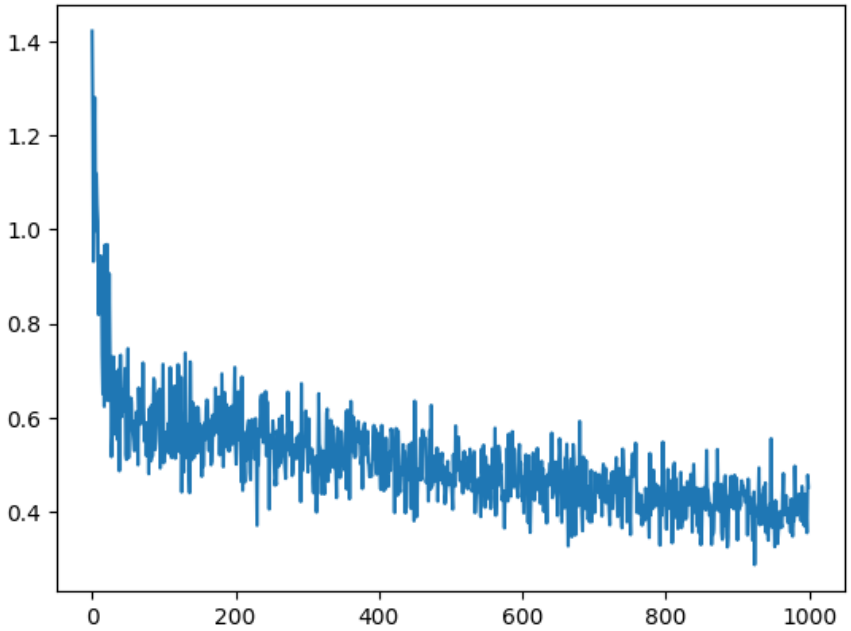
```
Loss = []
epochs = 1000
learning_rate = 0.01
batch_size = 50
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(1.4217163, 0.2871933)
```

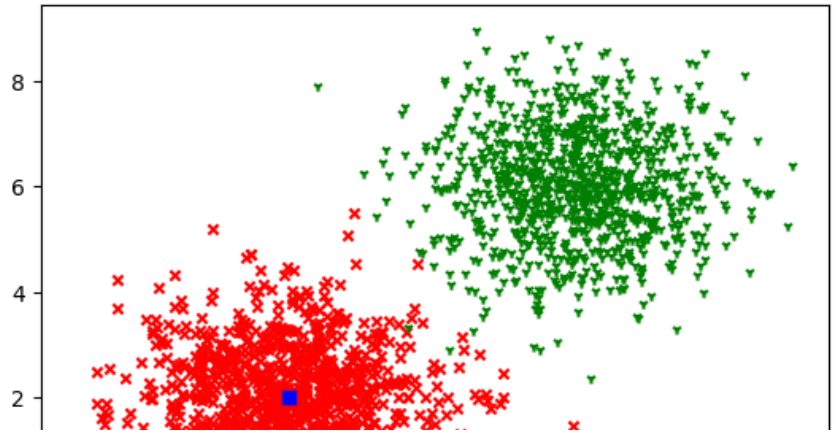
```
print(a.numpy())
print(b.numpy())
```

```
-0.112810634
0.5943694
```

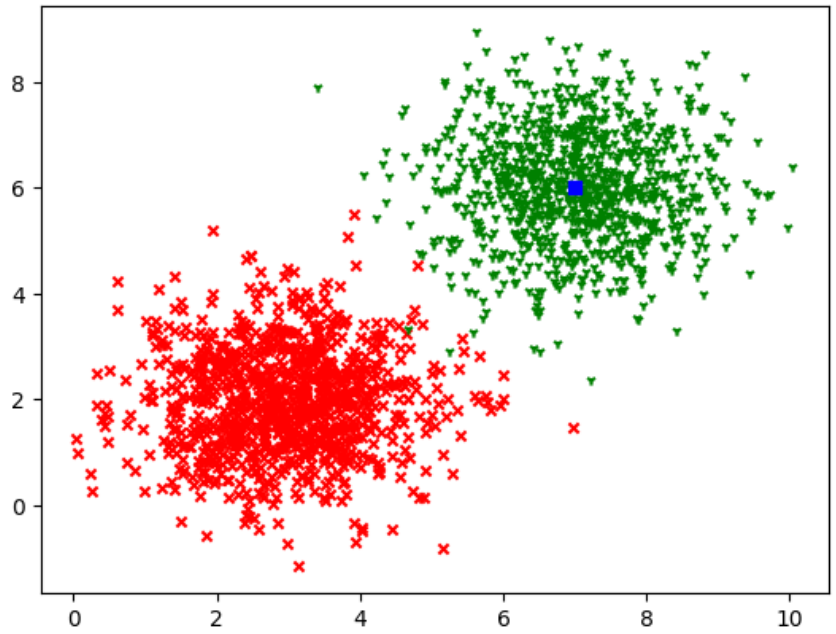
```
plt.plot(Loss)
plt.show()
```



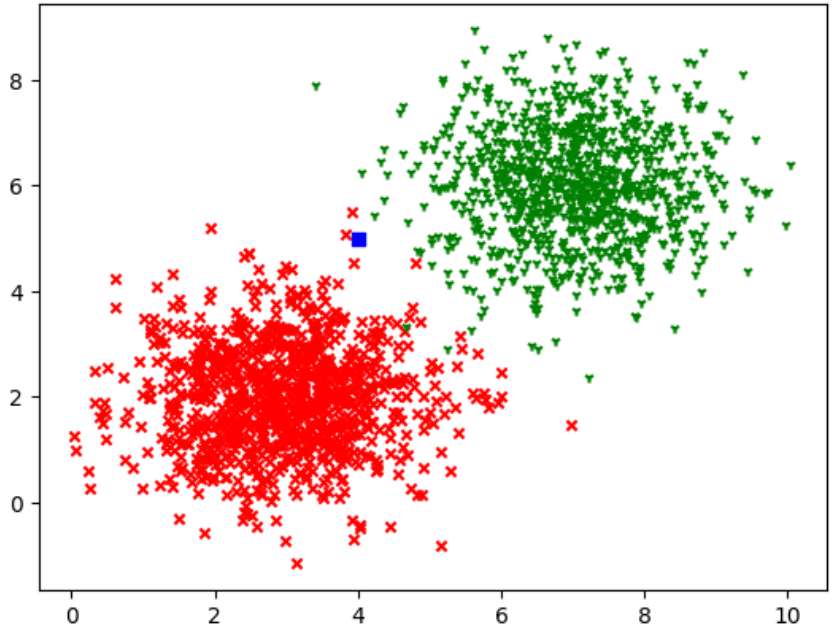

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Batch size - 20

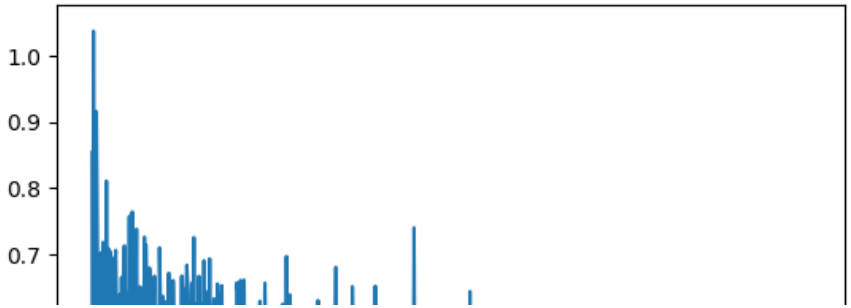
```
Loss = []
epochs = 1000
learning_rate = 0.01
batch_size = 20
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(1.0369635, 0.23119922)
```

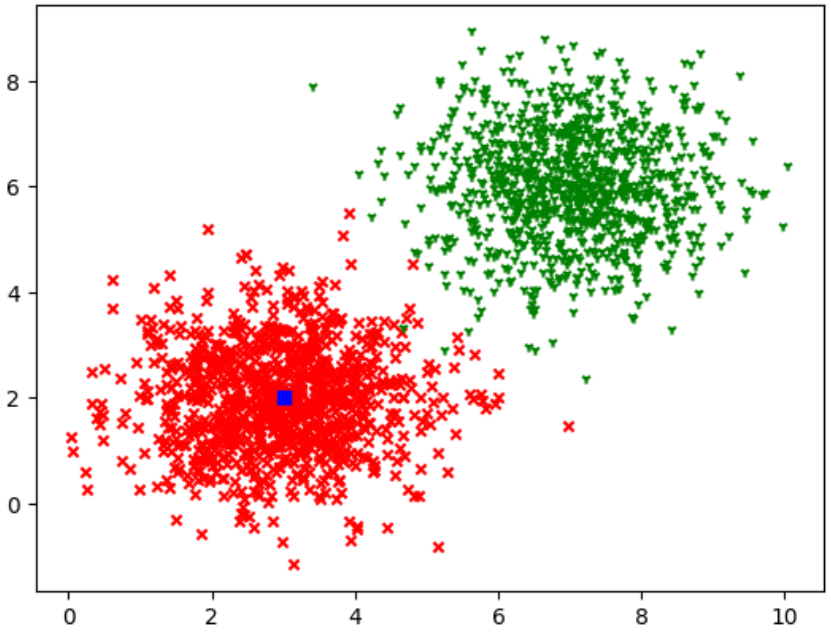
```
print(a.numpy())
print(b.numpy())
```

```
-0.025181422
0.518855
```

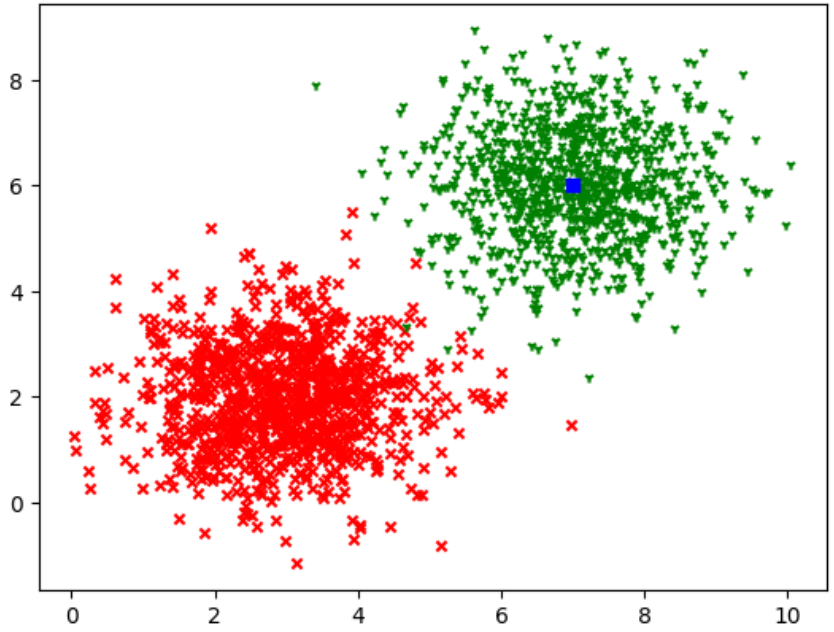
```
plt.plot(Loss)
plt.show()
```



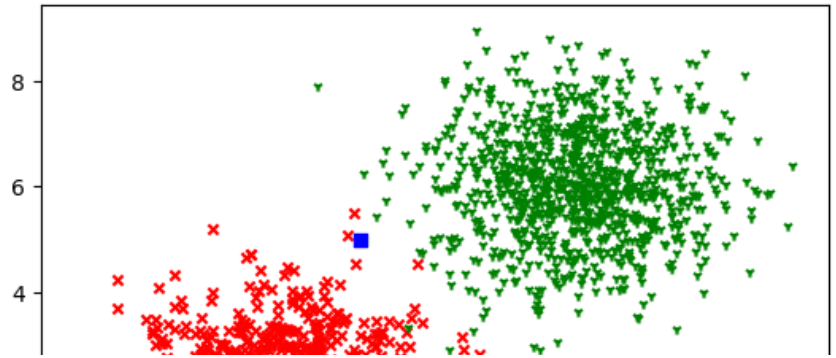

```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Batch size - 100

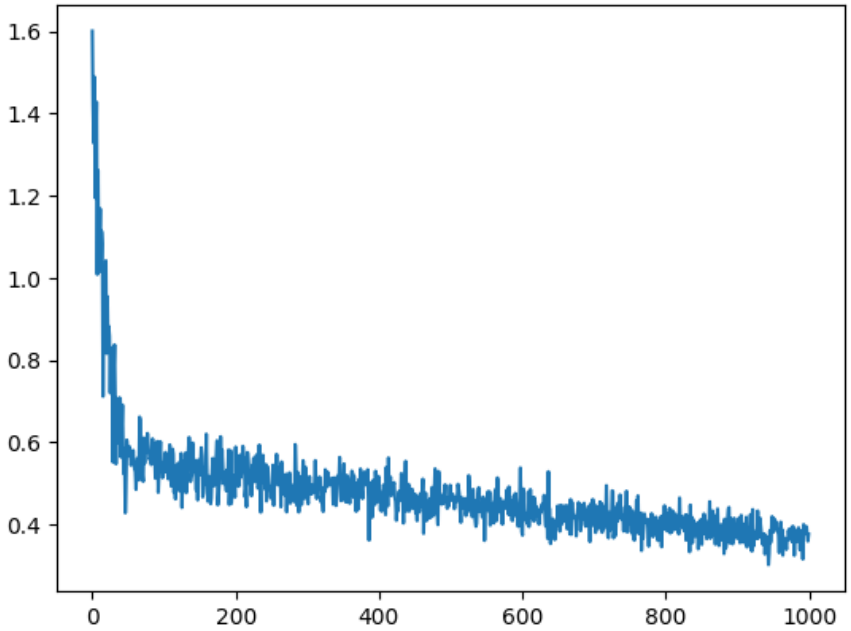
```
Loss = []
epochs = 1000
learning_rate = 0.01
batch_size = 100
a = tf.Variable(random.random())
b = tf.Variable(random.random())
c = tf.Variable(random.random())
for _ in range(epochs):
  xs_batch,ys_batch,labels_batch = subset_dataset(xs,ys,labels,batch_size)
  with tf.GradientTape() as tape:
    pred_l = tf.sigmoid(a * xs_batch + b * ys_batch + c)
    #print(label_batch.shape)
    loss = loss_fn_grad(labels_batch, pred_l)
    Loss.append(loss.numpy())

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

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


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

```
(1.600503, 0.3014175)
```

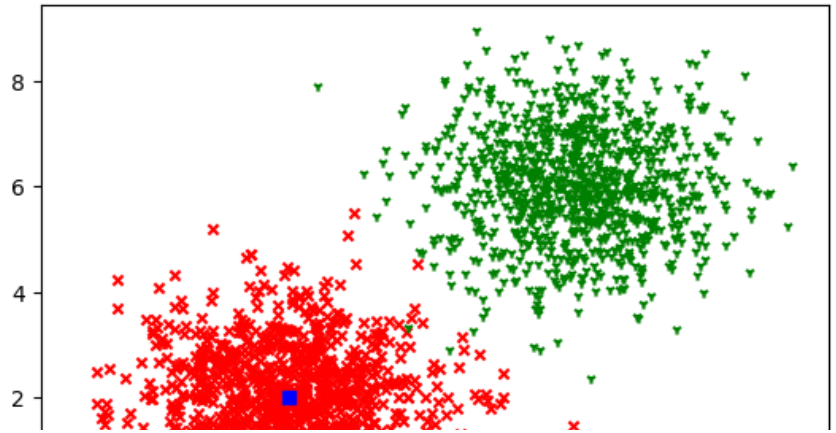
```
print(a.numpy())
print(b.numpy())
```

```
-0.09112978
0.61334753
```

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




```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```