Exercise 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
```

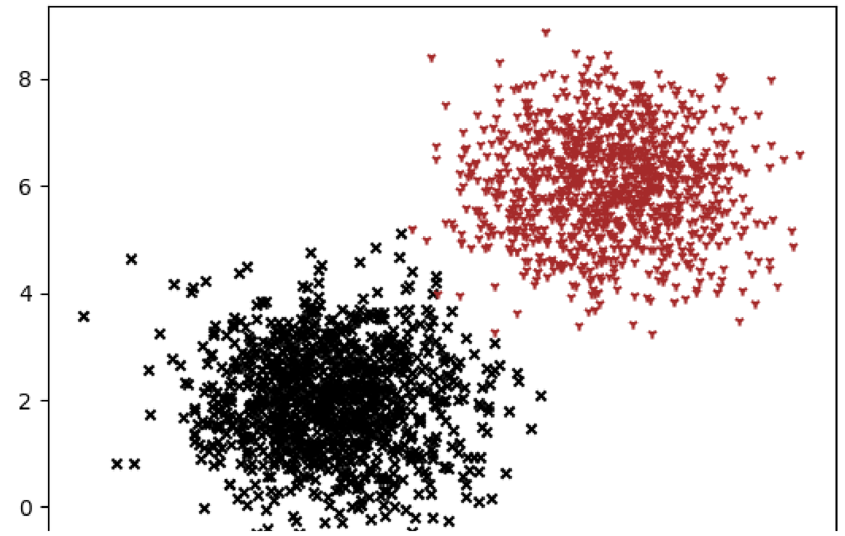
```
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
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
```

```
           2.76857777,  3.05580089,  1.46455197,  4.30984666,  4.0081525 ,
           3.30948566,  2.65892936,  4.54708041,  3.3536456 ,  2.64132637,
           1.91270728,  2.28289622,  1.98510396,  4.50041241,  3.02702409,
           3.43281255,  3.09853993,  3.16128403,  3.62381545,  3.31061833,
           3.10332631,  4.65475797,  2.82132597,  5.55659173,  0.1992336 ,
           3.3286668 ,  2.30486388,  2.1906411 ,  1.23671327,  2.61964142,
           2.61999692,  2.45492048,  4.9837424 ,  1.67535553,  2.63047348,
           3.73767474,  4.0986719 ,  2.68707446,  3.62350634,  1.45070148,
           4.52935102,  4.09046056,  3.45308144,  2.75901958,  2.39938623,
           2.70931953,  3.34270314,  2.26173305,  3.81247003,  2.74178278,
           3.36031086,  4.60967789,  2.86390102,  4.19866739,  2.27490456,
           4.43047177,  2.03006323,  3.10469332,  2.79190053,  3.19264955,
           2.08360951,  4.65623311,  1.27400255,  3.2272532 ,  2.64194927,
           1.89099932,  1.38163391,  3.73287552,  3.62950425,  3.7473768 ,
           3.53349219,  1.66943243,  4.18337928,  2.25480363,  2.37689968,
           1.63581301,  3.90752455,  3.93871394,  2.41022709,  3.36137868,
           4.74228693,  0.48701304,  2.73775202,  3.55490039,  2.36108197,
           4.97994702,  4.54542903,  3.72602175,  2.78391812,  3.77055318,
           3.3955416 ,  2.67375828,  2.25612879,  1.56181916,  3.76082979,
           4.3261713 ,  3.36876662,  2.69526173,  1.90672271,  3.18184957,
           2.3843636 ,  2.99288444,  2.65162821,  2.49780499,  3.79412158,
           3.17941147,  2.98437068,  2.39272109,  5.202545  ,  3.44023152,
           3.63917235,  3.00954373,  1.9515941 ,  2.43108033,  2.09541243,
           1.88771844,  2.45698092,  2.57063964,  3.91299906,  2.97756476,
           2.98083935,  4.69267141,  2.39218397,  1.70003808,  3.94808565,
           2.67958261,  3.47190478,  2.48356196,  3.16802027,  1.78518477,
           2.80122073,  4.47161755,  2.72641721,  3.35765132,  3.65103761,
           2.20791223,  2.32981877,  3.45861151,  1.45496228,  5.3351428 ,
           4.7084647 ,  1.78428851,  3.59957314,  2.56614516,  3.19995689])
```

```python
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))
```

```python
def split_dataset(data_points, label,subset_size):
    arr = np.arange(len(data_points))
    l=len(data_points)
    s=int(subset_size*l)
    np.random.shuffle(arr)
    data_points_val =data_points[arr[0:s]]
    label_val = label[arr[0:s]]
    data_points_train = data_points[arr[:int(l*(1-subset_size))]]
    label_train = label[arr[:int(l*(1-subset_size))]]

    return data_points_train,label_train,data_points_val,label_val
```

```
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


def subset_dataset_concatenated(data,label,subset_size):
    arr = np.arange(len(data))
    np.random.shuffle(arr)
    data_train = data[arr[0:subset_size]]
    label_train = label[arr[0:subset_size]]
    return data_train,label_train
```

```
labels.shape
```

```
(2000,)
```

```
Loss = []
Val_loss = []
epochs = 5000
learning_rate = 0.1
batch_size = 20

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
    loss tf.Tensor(0.5122043154210?, shape=(), dtype=float64) val_loss tf.Tensor(0.5168481221810552, shape=(), dtype=float64)
    loss tf.Tensor(0.5256652933447883, shape=(), dtype=float64) val_loss tf.Tensor(0.5166535053015568, shape=(), dtype=float64)
    loss tf.Tensor(0.5178850405770185, shape=(), dtype=float64) val_loss tf.Tensor(0.5203052381603911, shape=(), dtype=float64)
    loss tf.Tensor(0.517922249328594, shape=(), dtype=float64) val_loss tf.Tensor(0.5154060740405815, shape=(), dtype=float64)
    loss tf.Tensor(0.5152846908257349, shape=(), dtype=float64) val_loss tf.Tensor(0.5111547236876955, shape=(), dtype=float64)
    loss tf.Tensor(0.5179300538121787, shape=(), dtype=float64) val_loss tf.Tensor(0.5142287949998849, shape=(), dtype=float64)
    loss tf.Tensor(0.512815642658637, shape=(), dtype=float64) val_loss tf.Tensor(0.5183142711873845, shape=(), dtype=float64)
    loss tf.Tensor(0.5147071733618589, shape=(), dtype=float64) val_loss tf.Tensor(0.5195010178465658, shape=(), dtype=float64)
    loss tf.Tensor(0.517624940687482, shape=(), dtype=float64) val_loss tf.Tensor(0.5092411047700509, shape=(), dtype=float64)
    loss tf.Tensor(0.5148768149278122, shape=(), dtype=float64) val_loss tf.Tensor(0.5097451641217741, shape=(), dtype=float64)
    loss tf.Tensor(0.512340229615924７, shape=(), dtype=float64) val_loss tf.Tensor(0.5242343871866244, shape=(), dtype=float64)
    loss tf.Tensor(0.5158228739727162, shape=(), dtype=float64) val_loss tf.Tensor(0.5176552019419643, shape=(), dtype=float64)
    loss tf.Tensor(0.5181808717704712, shape=(), dtype=float64) val_loss tf.Tensor(0.5302942459486595, shape=(), dtype=float64)
    loss tf.Tensor(0.5210640558914632, shape=(), dtype=float64) val_loss tf.Tensor(0.5091564510948471, shape=(), dtype=float64)
    loss tf.Tensor(0.5196437340793592, shape=(), dtype=float64) val_loss tf.Tensor(0.5108230031725508, shape=(), dtype=float64)
    loss tf.Tensor(0.5101729655144, shape=(), dtype=float64) val_loss tf.Tensor(0.5100541946290931, shape=(), dtype=float64)
    loss tf.Tensor(0.5073725707414354, shape=(), dtype=float64) val_loss tf.Tensor(0.5256094437149509, shape=(), dtype=float64)
    loss tf.Tensor(0.5140487890774331, shape=(), dtype=float64) val_loss tf.Tensor(0.5185798935276333, shape=(), dtype=float64)
    loss tf.Tensor(0.5147100753424215, shape=(), dtype=float64) val_loss tf.Tensor(0.5126580177094191, shape=(), dtype=float64)
    loss tf.Tensor(0.512593865203139, shape=(), dtype=float64) val_loss tf.Tensor(0.523060146363972, shape=(), dtype=float64)
    loss tf.Tensor(0.5165768522340816, shape=(), dtype=float64) val_loss tf.Tensor(0.5215936957953248, shape=(), dtype=float64)
    loss tf.Tensor(0.5145151858330945, shape=(), dtype=float64) val_loss tf.Tensor(0.5132791642210766, shape=(), dtype=float64)
    loss tf.Tensor(0.511823165370401, shape=(), dtype=float64) val_loss tf.Tensor(0.5165108258351669, shape=(), dtype=float64)
    loss tf.Tensor(0.5340399098517743, shape=(), dtype=float64) val_loss tf.Tensor(0.5075444861725433, shape=(), dtype=float64)
    loss tf.Tensor(0.5169254816341875, shape=(), dtype=float64) val_loss tf.Tensor(0.5119881931845672, shape=(), dtype=float64)
    loss tf.Tensor(0.5210333395143512, shape=(), dtype=float64) val_loss tf.Tensor(0.5140113103480203, shape=(), dtype=float64)
    loss tf.Tensor(0.515202840404932, shape=(), dtype=float64) val_loss tf.Tensor(0.512332780709589, shape=(), dtype=float64)
    loss tf.Tensor(0.5150698649983365, shape=(), dtype=float64) val_loss tf.Tensor(0.5120866121899313, shape=(), dtype=float64)
    loss tf.Tensor(0.5304406833370937, shape=(), dtype=float64) val_loss tf.Tensor(0.5156165411408342, shape=(), dtype=float64)
    loss tf.Tensor(0.5159516568757614, shape=(), dtype=float64) val_loss tf.Tensor(0.5313480072067179, shape=(), dtype=float64)
    loss tf.Tensor(0.527278300045394, shape=(), dtype=float64) val_loss tf.Tensor(0.5242581866241394, shape=(), dtype=float64)
    loss tf.Tensor(0.5300226215717415, shape=(), dtype=float64) val_loss tf.Tensor(0.5203665836874464, shape=(), dtype=float64)
    loss tf.Tensor(0.5162360599561746, shape=(), dtype=float64) val_loss tf.Tensor(0.5132870609061891, shape=(), dtype=float64)
    loss tf.Tensor(0.5348840565998073, shape=(), dtype=float64) val_loss tf.Tensor(0.5116777943532278, shape=(), dtype=float64)
    loss tf.Tensor(0.5213577937715219, shape=(), dtype=float64) val_loss tf.Tensor(0.5092627719822975, shape=(), dtype=float64)
    loss tf.Tensor(0.5102712250713457, shape=(), dtype=float64) val_loss tf.Tensor(0.5122154219339361, shape=(), dtype=float64)
    loss tf.Tensor(0.5192766449738351, shape=(), dtype=float64) val_loss tf.Tensor(0.5203141652220289, shape=(), dtype=float64)
    loss tf.Tensor(0.5133585054523104, shape=(), dtype=float64) val_loss tf.Tensor(0.5112405492728064, shape=(), dtype=float64)
    loss tf.Tensor(0.5161412473586546, shape=(), dtype=float64) val_loss tf.Tensor(0.5224799881012637, shape=(), dtype=float64)
    loss tf.Tensor(0.5114006490589237, shape=(), dtype=float64) val_loss tf.Tensor(0.5136758197089615, shape=(), dtype=float64)
    loss tf.Tensor(0.508903866092442, shape=(), dtype=float64) val_loss tf.Tensor(0.5127944070751355, shape=(), dtype=float64)
    loss tf.Tensor(0.5321286128639814, shape=(), dtype=float64) val_loss tf.Tensor(0.5152197623047291, shape=(), dtype=float64)
    loss tf.Tensor(0.5126511748973185, shape=(), dtype=float64) val_loss tf.Tensor(0.5119886462276876, shape=(), dtype=float64)
    loss tf.Tensor(0.5115058768604321, shape=(), dtype=float64) val_loss tf.Tensor(0.5270552301924957, shape=(), dtype=float64)
    loss tf.Tensor(0.5148654887241977, shape=(), dtype=float64) val_loss tf.Tensor(0.5129883036155126, shape=(), dtype=float64)
    loss tf.Tensor(0.5125567617774643, shape=(), dtype=float64) val_loss tf.Tensor(0.514186096690384, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
    3200 1600 800 400
```

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

```
(0.8669451729799297, 0.5066340502394049)
```

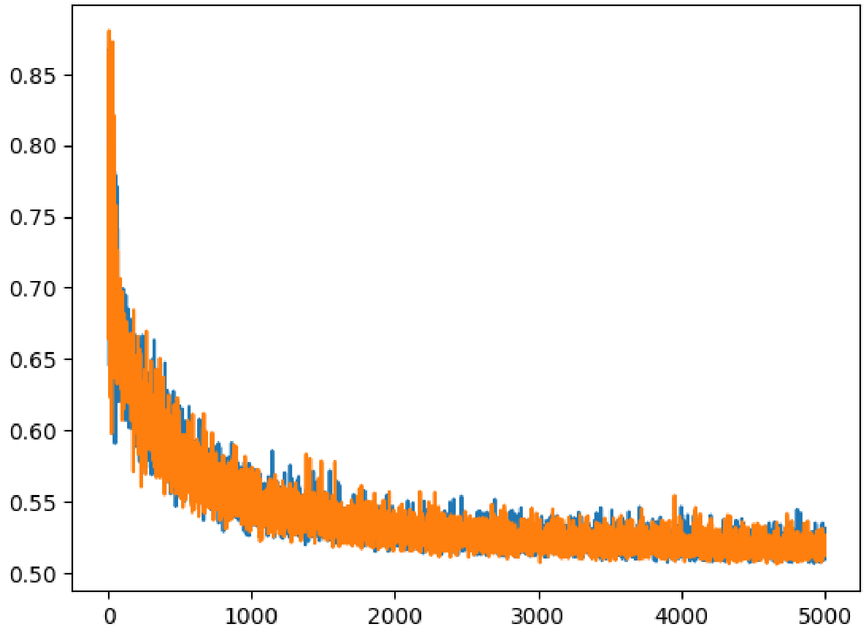
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8803401434875809, 0.5061144178887617)
```

```
print(w.numpy())
print(b.numpy())
```

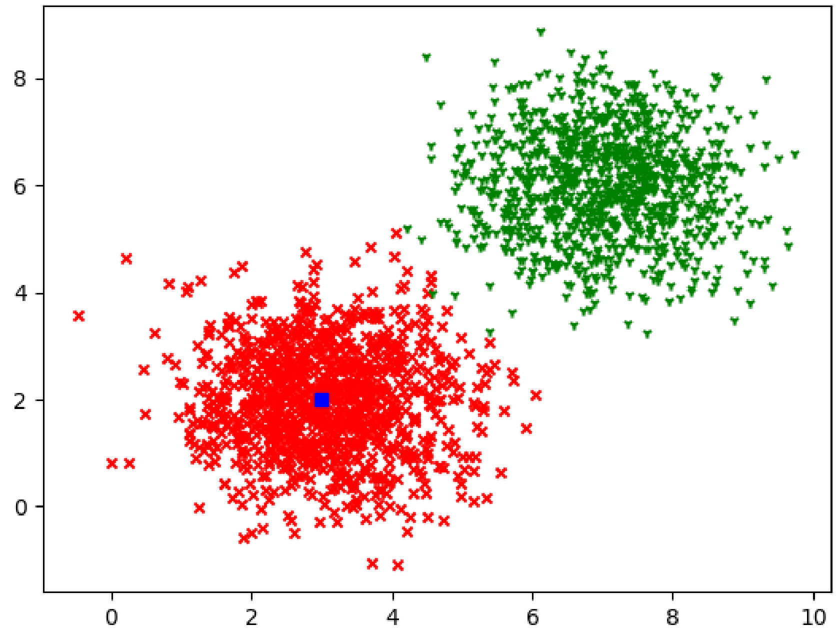
```
[[ 0.12513583  1.02181097]
 [-0.29873544  1.00025091]]
[ 5.4495681  -3.89044614]
```

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

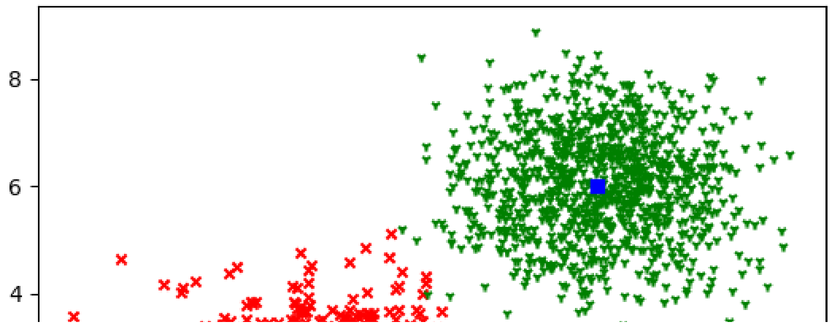


```
x=3.0
```

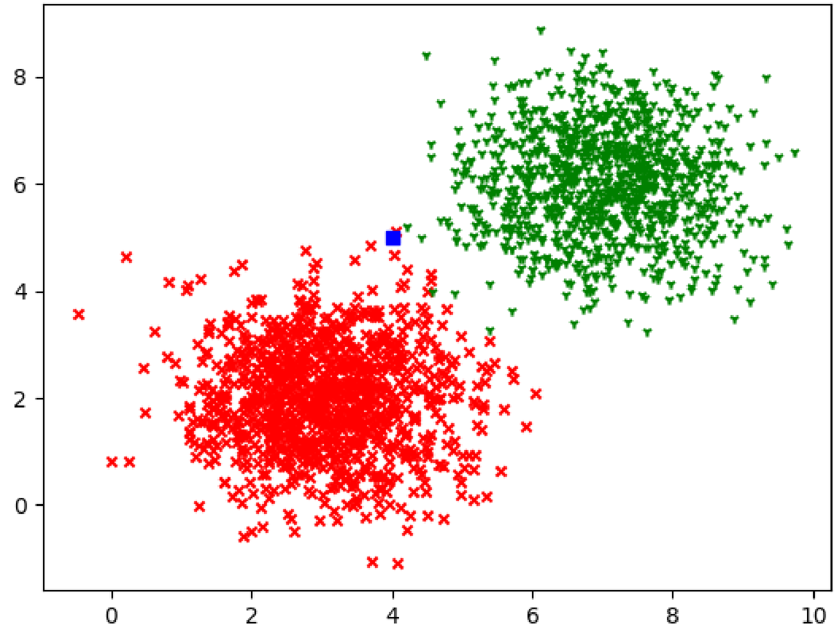
```
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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 0.01

```
Loss = []
Val_loss = []
epochs = 5000
learning_rate = 0.01
batch_size = 20

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
     loss tf.Tensor(0.729989115084902, shape=(), dtype=float64) val_loss tf.Tensor(0.755068289528069 4, shape=(), dtype=float64)
     loss tf.Tensor(0.803648987884461, shape=(), dtype=float64) val_loss tf.Tensor(0.6548308088420781, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
     3200 1600 800 400
```

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

```
     (0.9331220401407656, 0.556582217420378)
```

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

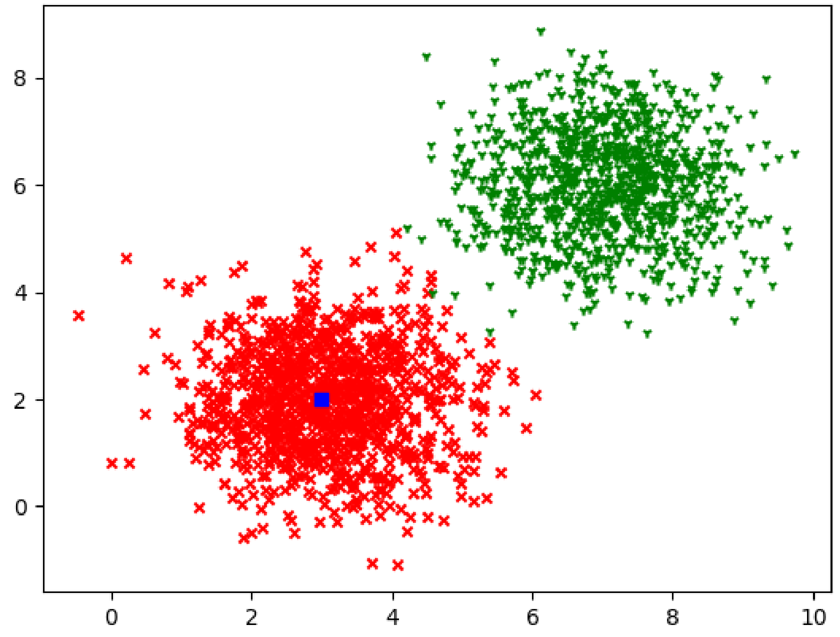
```
     (1.003188993512679, 0.5808985670668569)
```

```
print(w.numpy())
print(b.numpy())
```

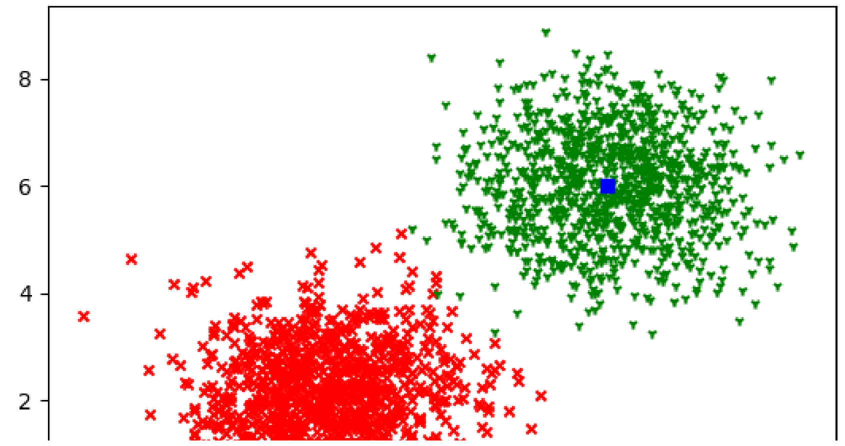
```
     [[1.25592541 0.31700005]
      [1.21573953 0.16564472]]
     [1.05052379 0.08972305]
```

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

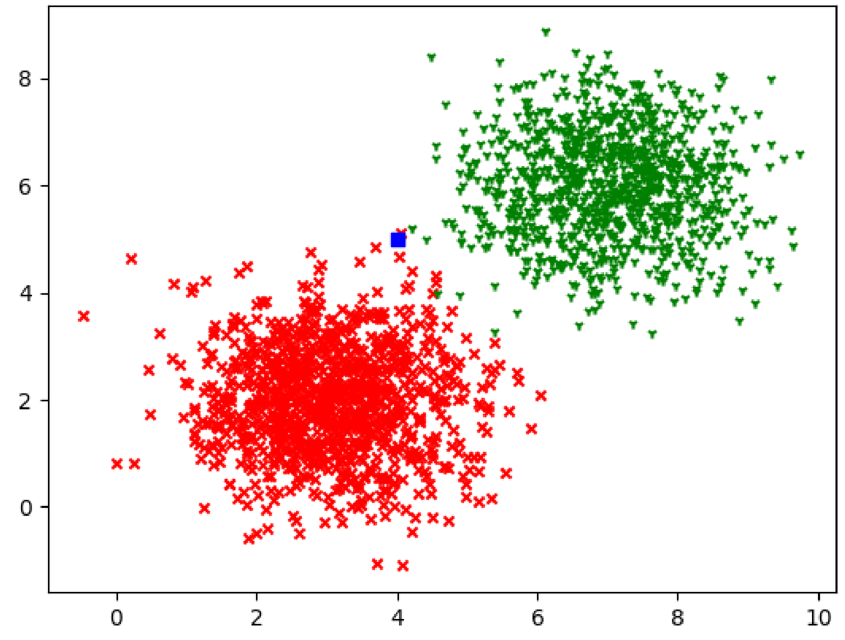
```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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 = []
Val_loss = []
epochs = 5000
learning_rate = 0.1
batch_size = 20

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
loss tf.Tensor(0.5273087539411385, shape=(), dtype=float64) val_loss tf.Tensor(0.5177295795368652, shape=(), dtype=float64)
loss tf.Tensor(0.5189430605543036, shape=(), dtype=float64) val_loss tf.Tensor(0.5155052180067216, shape=(), dtype=float64)
loss tf.Tensor(0.513511756142648, shape=(), dtype=float64) val_loss tf.Tensor(0.5127919148822238, shape=(), dtype=float64)
loss tf.Tensor(0.5101764309597876, shape=(), dtype=float64) val_loss tf.Tensor(0.5189958658493179, shape=(), dtype=float64)
loss tf.Tensor(0.5196424005534986, shape=(), dtype=float64) val_loss tf.Tensor(0.5118618637521382, shape=(), dtype=float64)
loss tf.Tensor(0.5188501951362374, shape=(), dtype=float64) val_loss tf.Tensor(0.5112260413259253, shape=(), dtype=float64)
loss tf.Tensor(0.5209071051856041, shape=(), dtype=float64) val_loss tf.Tensor(0.5133233718033687, shape=(), dtype=float64)
loss tf.Tensor(0.5078124238422042, shape=(), dtype=float64) val_loss tf.Tensor(0.513196475979453, shape=(), dtype=float64)
loss tf.Tensor(0.5197721386633857, shape=(), dtype=float64) val_loss tf.Tensor(0.5177385955155718, shape=(), dtype=float64)
loss tf.Tensor(0.5146268990527558, shape=(), dtype=float64) val_loss tf.Tensor(0.5103813188940525, shape=(), dtype=float64)
loss tf.Tensor(0.5085737981766647, shape=(), dtype=float64) val_loss tf.Tensor(0.5116532543527562, shape=(), dtype=float64)
loss tf.Tensor(0.5183905189085314, shape=(), dtype=float64) val_loss tf.Tensor(0.524131139021734, shape=(), dtype=float64)
loss tf.Tensor(0.5195840583549654, shape=(), dtype=float64) val_loss tf.Tensor(0.5099307892047982, shape=(), dtype=float64)
loss tf.Tensor(0.5174512131164086, shape=(), dtype=float64) val_loss tf.Tensor(0.5132162402928945, shape=(), dtype=float64)
loss tf.Tensor(0.5191790627562589, shape=(), dtype=float64) val_loss tf.Tensor(0.5125322726205608, shape=(), dtype=float64)
loss tf.Tensor(0.5253541363078215, shape=(), dtype=float64) val_loss tf.Tensor(0.5103687762148537, shape=(), dtype=float64)
loss tf.Tensor(0.5137894618497632, shape=(), dtype=float64) val_loss tf.Tensor(0.5126329741052267, shape=(), dtype=float64)
loss tf.Tensor(0.5109347437264484, shape=(), dtype=float64) val_loss tf.Tensor(0.5111489410830776, shape=(), dtype=float64)
loss tf.Tensor(0.511982794179852, shape=(), dtype=float64) val_loss tf.Tensor(0.5198977272035724, shape=(), dtype=float64)
loss tf.Tensor(0.5177734536616434, shape=(), dtype=float64) val_loss tf.Tensor(0.5106665128852244, shape=(), dtype=float64)
loss tf.Tensor(0.5116969042983642, shape=(), dtype=float64) val_loss tf.Tensor(0.5165137157732602, shape=(), dtype=float64)
loss tf.Tensor(0.5092038370077286, shape=(), dtype=float64) val_loss tf.Tensor(0.5214779758534336, shape=(), dtype=float64)
loss tf.Tensor(0.5103259937873792, shape=(), dtype=float64) val_loss tf.Tensor(0.5137889764512019, shape=(), dtype=float64)
loss tf.Tensor(0.521585317213763, shape=(), dtype=float64) val_loss tf.Tensor(0.5139719993535377, shape=(), dtype=float64)
loss tf.Tensor(0.5139860644674012, shape=(), dtype=float64) val_loss tf.Tensor(0.5091751884853066, shape=(), dtype=float64)
loss tf.Tensor(0.5194482426706601, shape=(), dtype=float64) val_loss tf.Tensor(0.5142152177471087, shape=(), dtype=float64)
loss tf.Tensor(0.5277311523814807, shape=(), dtype=float64) val_loss tf.Tensor(0.516572866145303, shape=(), dtype=float64)
loss tf.Tensor(0.5275317581187624, shape=(), dtype=float64) val_loss tf.Tensor(0.5163296244159018, shape=(), dtype=float64)
loss tf.Tensor(0.5165166439724884, shape=(), dtype=float64) val_loss tf.Tensor(0.5216401444847845, shape=(), dtype=float64)
loss tf.Tensor(0.5222676220368603, shape=(), dtype=float64) val_loss tf.Tensor(0.5165763816909597, shape=(), dtype=float64)
loss tf.Tensor(0.529621248932491, shape=(), dtype=float64) val_loss tf.Tensor(0.5159660732874464, shape=(), dtype=float64)
loss tf.Tensor(0.523467165584597, shape=(), dtype=float64) val_loss tf.Tensor(0.5112844699205696, shape=(), dtype=float64)
loss tf.Tensor(0.5167164293050123, shape=(), dtype=float64) val_loss tf.Tensor(0.5162130818230658, shape=(), dtype=float64)
loss tf.Tensor(0.5146041560776438, shape=(), dtype=float64) val_loss tf.Tensor(0.512288435507049, shape=(), dtype=float64)
loss tf.Tensor(0.5153606554209712, shape=(), dtype=float64) val_loss tf.Tensor(0.514768528786061, shape=(), dtype=float64)
loss tf.Tensor(0.5127451097266329, shape=(), dtype=float64) val_loss tf.Tensor(0.5111088140193374, shape=(), dtype=float64)
loss tf.Tensor(0.5177285492384475, shape=(), dtype=float64) val_loss tf.Tensor(0.5085369618061211, shape=(), dtype=float64)
loss tf.Tensor(0.5157969115330815, shape=(), dtype=float64) val_loss tf.Tensor(0.5170849031067417, shape=(), dtype=float64)
loss tf.Tensor(0.5200538685976491, shape=(), dtype=float64) val_loss tf.Tensor(0.5110168127815878, shape=(), dtype=float64)
loss tf.Tensor(0.5155814958632463, shape=(), dtype=float64) val_loss tf.Tensor(0.5118421676207223, shape=(), dtype=float64)
loss tf.Tensor(0.5137902806278498, shape=(), dtype=float64) val_loss tf.Tensor(0.5096147667692611, shape=(), dtype=float64)
loss tf.Tensor(0.5202994927257037, shape=(), dtype=float64) val_loss tf.Tensor(0.510025462480552, shape=(), dtype=float64)
loss tf.Tensor(0.516016810108774, shape=(), dtype=float64) val_loss tf.Tensor(0.5161654938882435, shape=(), dtype=float64)
loss tf.Tensor(0.5169116659842842, shape=(), dtype=float64) val_loss tf.Tensor(0.5173557271158268, shape=(), dtype=float64)
loss tf.Tensor(0.510662568305275, shape=(), dtype=float64) val_loss tf.Tensor(0.5111941127459524, shape=(), dtype=float64)
loss tf.Tensor(0.513524858127526, shape=(), dtype=float64) val_loss tf.Tensor(0.5129694240868237, shape=(), dtype=float64)
loss tf.Tensor(0.5217809560686683, shape=(), dtype=float64) val_loss tf.Tensor(0.515749584229652, shape=(), dtype=float64)
loss tf.Tensor(0.5146141541539442, shape=(), dtype=float64) val_loss tf.Tensor(0.5193874062781482, shape=(), dtype=float64)
loss tf.Tensor(0.521246963600418, shape=(), dtype=float64) val_loss tf.Tensor(0.5110375766913942, shape=(), dtype=float64)
loss tf.Tensor(0.5123742983787526, shape=(), dtype=float64) val_loss tf.Tensor(0.5133331853144293, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
3200 1600 800 400
```

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

```
(0.7882050912372996, 0.5058717040309512)
```

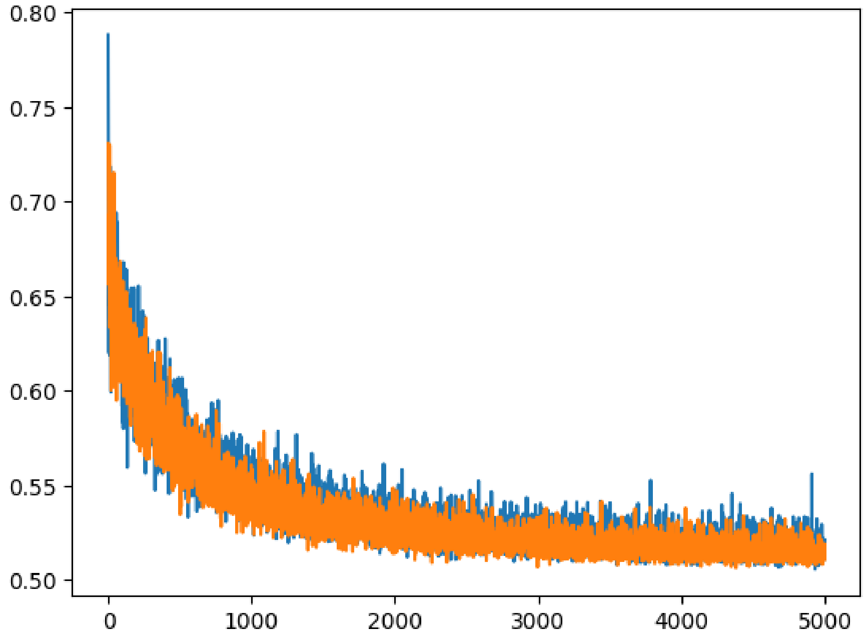
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.7308242559864719, 0.5065688514518036)
```

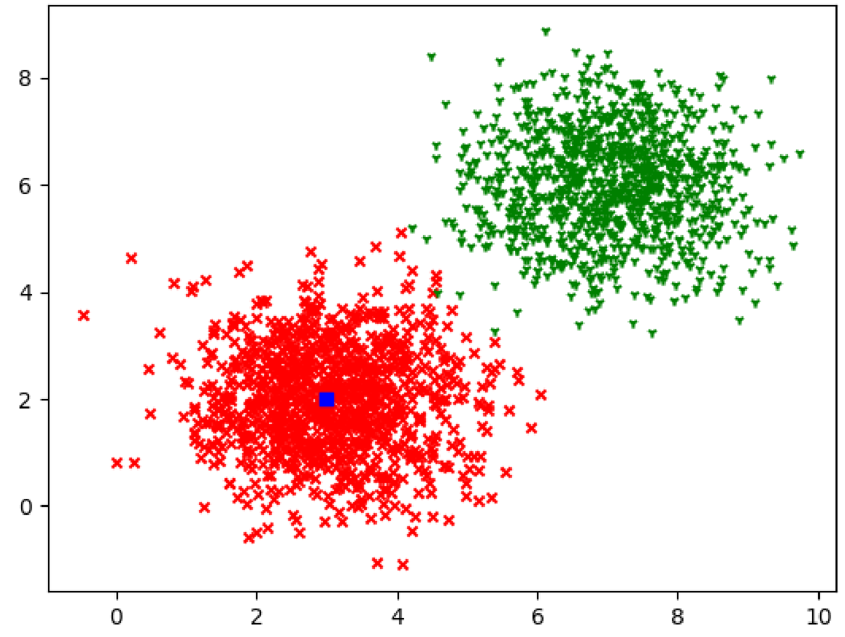
```
print(w.numpy())
print(b.numpy())
```

```
[[-0.29598607  0.5959906 ]
 [-0.26613405  1.01544346]]
[ 5.14409557 -4.2707041 ]
```

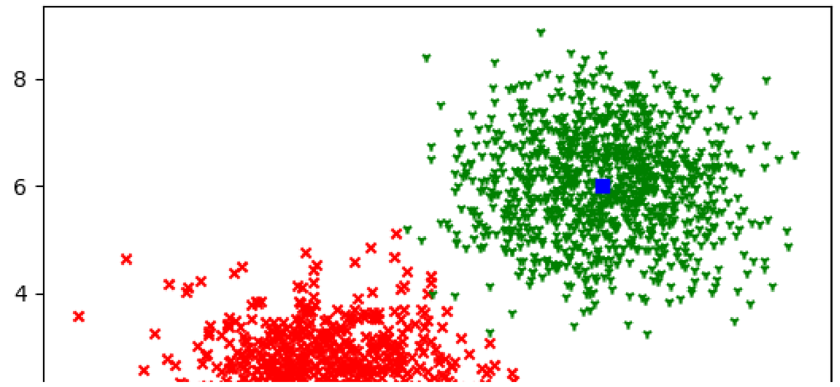
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



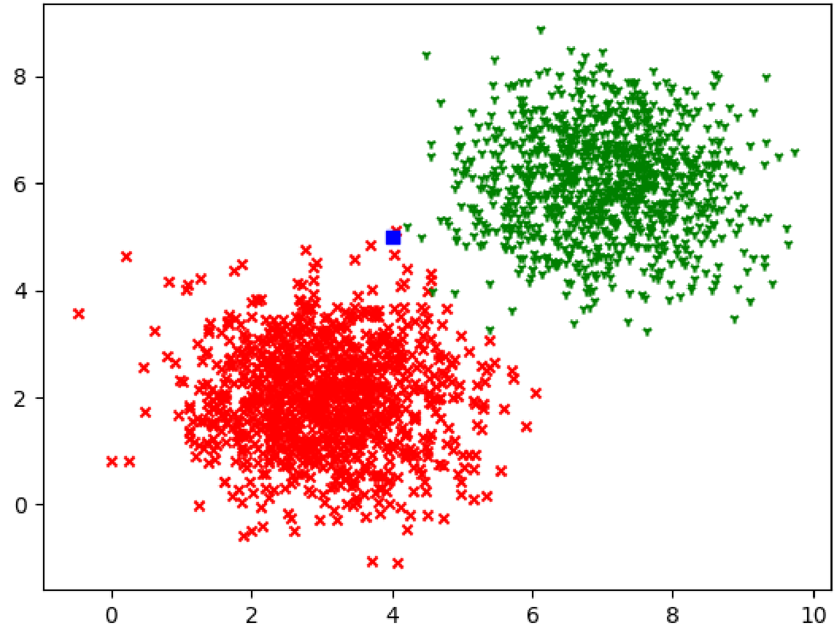
```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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 = []
Val_loss = []
epochs = 100
learning_rate = 0.1
batch_size = 20

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
        loss tf.Tensor(0.6733957491870834, shape=(), dtype=float64) val_loss tf.Tensor(0.6671950717140401, shape=(), dtype=float64)
        loss tf.Tensor(0.7246497654675836, shape=(), dtype=float64) val_loss tf.Tensor(0.6715672661755365, shape=(), dtype=float64)
        loss tf.Tensor(0.6609371367883302, shape=(), dtype=float64) val_loss tf.Tensor(0.6739284028778048, shape=(), dtype=float64)
        loss tf.Tensor(0.6736872222325825, shape=(), dtype=float64) val_loss tf.Tensor(0.7087440940967824, shape=(), dtype=float64)
        loss tf.Tensor(0.7445269305375025, shape=(), dtype=float64) val_loss tf.Tensor(0.6649892569953518, shape=(), dtype=float64)
        loss tf.Tensor(0.6687288281797377, shape=(), dtype=float64) val_loss tf.Tensor(0.7217815540626853, shape=(), dtype=float64)
        loss tf.Tensor(0.6320415847571409, shape=(), dtype=float64) val_loss tf.Tensor(0.7019695016165012, shape=(), dtype=float64)
        loss tf.Tensor(0.6746580162563881, shape=(), dtype=float64) val_loss tf.Tensor(0.6804945477783436, shape=(), dtype=float64)
        loss tf.Tensor(0.6643928763766408, shape=(), dtype=float64) val_loss tf.Tensor(0.7162736208076804, shape=(), dtype=float64)
        loss tf.Tensor(0.6719945606367692, shape=(), dtype=float64) val_loss tf.Tensor(0.6370117692853579, shape=(), dtype=float64)
        loss tf.Tensor(0.6639844689461608, shape=(), dtype=float64) val_loss tf.Tensor(0.6647068541250726, shape=(), dtype=float64)
        loss tf.Tensor(0.6804345397016153, shape=(), dtype=float64) val_loss tf.Tensor(0.7238732801657715, shape=(), dtype=float64)
        loss tf.Tensor(0.6827126150558348, shape=(), dtype=float64) val_loss tf.Tensor(0.7127205991655815, shape=(), dtype=float64)
        loss tf.Tensor(0.644492795199431, shape=(), dtype=float64) val_loss tf.Tensor(0.6920971734112922, shape=(), dtype=float64)
        loss tf.Tensor(0.6598345707786173, shape=(), dtype=float64) val_loss tf.Tensor(0.7152430086286412, shape=(), dtype=float64)
        loss tf.Tensor(0.6304325345283306, shape=(), dtype=float64) val_loss tf.Tensor(0.6545398400404258, shape=(), dtype=float64)
        loss tf.Tensor(0.6616881065420283, shape=(), dtype=float64) val_loss tf.Tensor(0.6858405164489018, shape=(), dtype=float64)
        loss tf.Tensor(0.620671449366859, shape=(), dtype=float64) val_loss tf.Tensor(0.7084741619312193, shape=(), dtype=float64)
        loss tf.Tensor(0.6664937455243646, shape=(), dtype=float64) val_loss tf.Tensor(0.6837972817732435, shape=(), dtype=float64)
        loss tf.Tensor(0.7101757866375109, shape=(), dtype=float64) val_loss tf.Tensor(0.7066420417021015, shape=(), dtype=float64)
        loss tf.Tensor(0.6992543087946675, shape=(), dtype=float64) val_loss tf.Tensor(0.676808101458747, shape=(), dtype=float64)
        loss tf.Tensor(0.648066246893346, shape=(), dtype=float64) val_loss tf.Tensor(0.6958530072446778, shape=(), dtype=float64)
        loss tf.Tensor(0.6754348190736564, shape=(), dtype=float64) val_loss tf.Tensor(0.6847898981616553, shape=(), dtype=float64)
        loss tf.Tensor(0.7249233730152513, shape=(), dtype=float64) val_loss tf.Tensor(0.698585324121069, shape=(), dtype=float64)
        loss tf.Tensor(0.626890140903593, shape=(), dtype=float64) val_loss tf.Tensor(0.6931068047203345, shape=(), dtype=float64)
        loss tf.Tensor(0.643984107658615, shape=(), dtype=float64) val_loss tf.Tensor(0.6918733068380882, shape=(), dtype=float64)
        loss tf.Tensor(0.6727321368212407, shape=(), dtype=float64) val_loss tf.Tensor(0.6415228835096695, shape=(), dtype=float64)
        loss tf.Tensor(0.6604965854778566, shape=(), dtype=float64) val_loss tf.Tensor(0.6750646103222412, shape=(), dtype=float64)
        loss tf.Tensor(0.6721383177676291, shape=(), dtype=float64) val_loss tf.Tensor(0.7031128630330268, shape=(), dtype=float64)
        loss tf.Tensor(0.6627174108184721, shape=(), dtype=float64) val_loss tf.Tensor(0.7051839350263664, shape=(), dtype=float64)
        loss tf.Tensor(0.7021666395689927, shape=(), dtype=float64) val_loss tf.Tensor(0.6698325299178501, shape=(), dtype=float64)
        loss tf.Tensor(0.710454896450808, shape=(), dtype=float64) val_loss tf.Tensor(0.6686069624585107, shape=(), dtype=float64)
        loss tf.Tensor(0.6647213582212509, shape=(), dtype=float64) val_loss tf.Tensor(0.6593381449068024, shape=(), dtype=float64)
        loss tf.Tensor(0.6369359593509291, shape=(), dtype=float64) val_loss tf.Tensor(0.6598384277985323, shape=(), dtype=float64)
        loss tf.Tensor(0.6192108211476626, shape=(), dtype=float64) val_loss tf.Tensor(0.6664981779756161, shape=(), dtype=float64)
        loss tf.Tensor(0.6869505253548251, shape=(), dtype=float64) val_loss tf.Tensor(0.6069132964204423, shape=(), dtype=float64)
        loss tf.Tensor(0.669179178976899, shape=(), dtype=float64) val_loss tf.Tensor(0.6858540379882376, shape=(), dtype=float64)
        loss tf.Tensor(0.6382145373617545, shape=(), dtype=float64) val_loss tf.Tensor(0.6861475467419955, shape=(), dtype=float64)
        loss tf.Tensor(0.6702241097493398, shape=(), dtype=float64) val_loss tf.Tensor(0.6910532454448759, shape=(), dtype=float64)
        loss tf.Tensor(0.6882342601982667, shape=(), dtype=float64) val_loss tf.Tensor(0.6716889833219419, shape=(), dtype=float64)
        loss tf.Tensor(0.6311002346575305, shape=(), dtype=float64) val_loss tf.Tensor(0.6742937294435515, shape=(), dtype=float64)
        loss tf.Tensor(0.630272645983877, shape=(), dtype=float64) val_loss tf.Tensor(0.6577118580876269, shape=(), dtype=float64)
        loss tf.Tensor(0.6809905752431893, shape=(), dtype=float64) val_loss tf.Tensor(0.7177009493918087, shape=(), dtype=float64)
        loss tf.Tensor(0.6808002550548535, shape=(), dtype=float64) val_loss tf.Tensor(0.6762249705740324, shape=(), dtype=float64)
        loss tf.Tensor(0.6254514362833008, shape=(), dtype=float64) val_loss tf.Tensor(0.6414428944930146, shape=(), dtype=float64)
        loss tf.Tensor(0.6615450971489165, shape=(), dtype=float64) val_loss tf.Tensor(0.6638778563990193, shape=(), dtype=float64)
        loss tf.Tensor(0.666173578166893, shape=(), dtype=float64) val_loss tf.Tensor(0.7030899125675601, shape=(), dtype=float64)
        loss tf.Tensor(0.6072781613437183, shape=(), dtype=float64) val_loss tf.Tensor(0.6808090278060256, shape=(), dtype=float64)
        loss tf.Tensor(0.6454217675241385, shape=(), dtype=float64) val_loss tf.Tensor(0.6973908345165002, shape=(), dtype=float64)
        loss tf.Tensor(0.6200864000254587, shape=(), dtype=float64) val_loss tf.Tensor(0.6516227430630405, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
3200 1600 800 400
```

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

```
(0.7878333454760538, 0.6072781613437183)
```

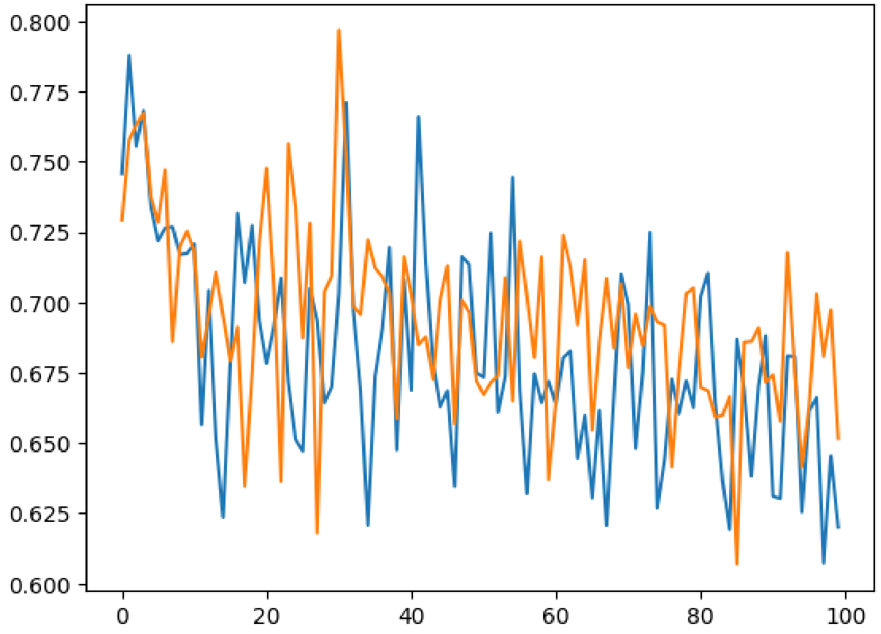
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.7969265178660555, 0.6069132964204423)
```

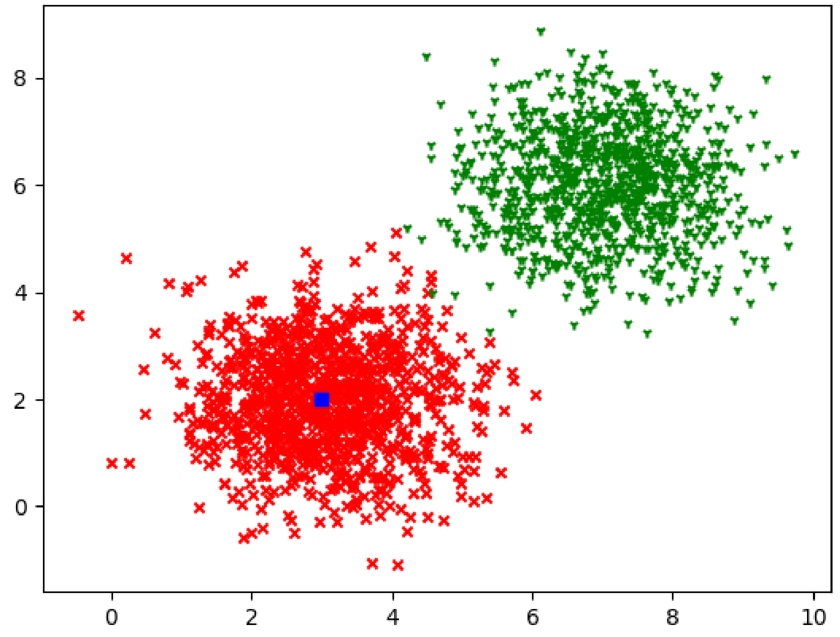
```
print(w.numpy())
print(b.numpy())
```

```
[[ 0.34865976  0.37357016]
 [-0.08901816  0.35148574]]
[0.93015622 0.49543372]
```

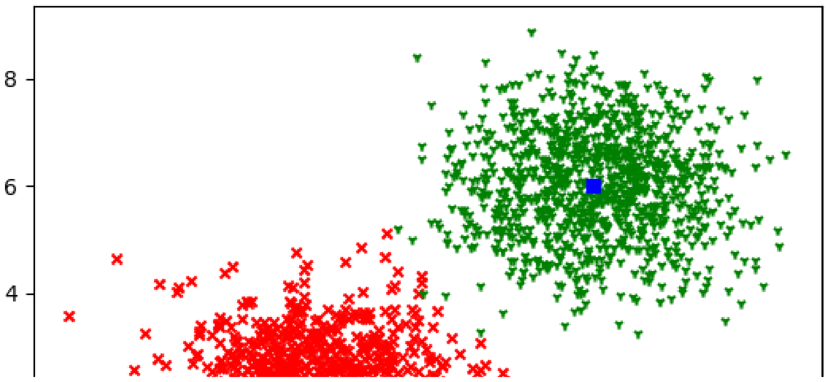
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



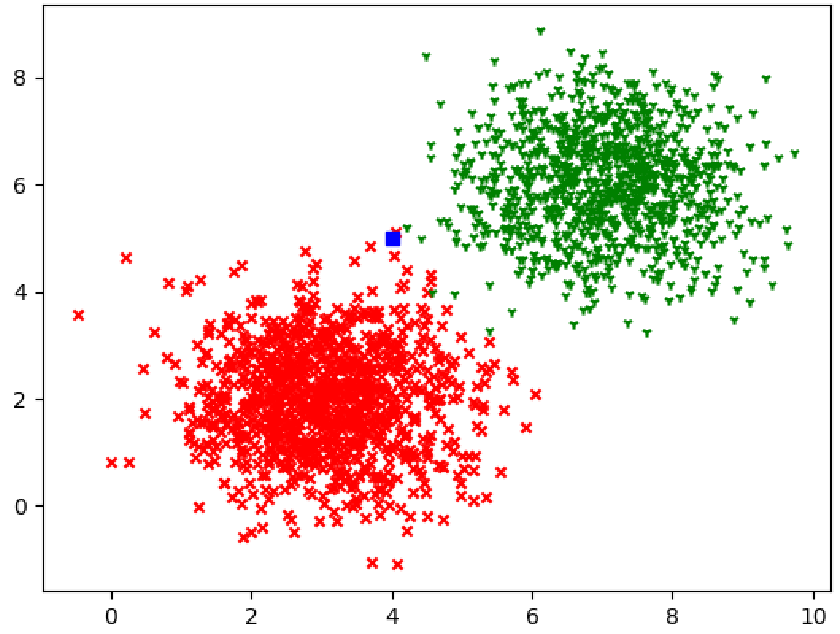
```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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 = []
Val_loss = []
epochs = 3000
learning_rate = 0.1
batch_size = 20

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
    loss tf.Tensor(0.8032663229449535, shape=(), dtype=float64) val_loss tf.Tensor(0.7533284721169359, shape=(), dtype=float64)
    loss tf.Tensor(0.8032780502616876, shape=(), dtype=float64) val_loss tf.Tensor(0.7533234681799102, shape=(), dtype=float64)
    loss tf.Tensor(0.7533483665090945, shape=(), dtype=float64) val_loss tf.Tensor(0.8035295460389478, shape=(), dtype=float64)
    loss tf.Tensor(0.7032946386007684, shape=(), dtype=float64) val_loss tf.Tensor(0.7532561658746768, shape=(), dtype=float64)
    loss tf.Tensor(0.7534063056010446, shape=(), dtype=float64) val_loss tf.Tensor(0.7535546572500454, shape=(), dtype=float64)
    loss tf.Tensor(0.6783898224984162, shape=(), dtype=float64) val_loss tf.Tensor(0.8038315174852343, shape=(), dtype=float64)
    loss tf.Tensor(0.803707072292880 3, shape=(), dtype=float64) val_loss tf.Tensor(0.7533070314333337, shape=(), dtype=float64)
    loss tf.Tensor(0.9032382688459677, shape=(), dtype=float64) val_loss tf.Tensor(0.7532743550626113, shape=(), dtype=float64)
    loss tf.Tensor(0.7782687492560763, shape=(), dtype=float64) val_loss tf.Tensor(0.7782871180483786, shape=(), dtype=float64)
    loss tf.Tensor(0.7533693808743397, shape=(), dtype=float64) val_loss tf.Tensor(0.7285701510759152, shape=(), dtype=float64)
    loss tf.Tensor(0.7056159927654255, shape=(), dtype=float64) val_loss tf.Tensor(0.8283929638248277, shape=(), dtype=float64)
    loss tf.Tensor(0.7034835643612104, shape=(), dtype=float64) val_loss tf.Tensor(0.7782697315447761, shape=(), dtype=float64)
    loss tf.Tensor(0.8033632907523938, shape=(), dtype=float64) val_loss tf.Tensor(0.7282472643100585, shape=(), dtype=float64)
    loss tf.Tensor(0.7533752060699157, shape=(), dtype=float64) val_loss tf.Tensor(0.8032764687065408, shape=(), dtype=float64)
    loss tf.Tensor(0.7783331154594265, shape=(), dtype=float64) val_loss tf.Tensor(0.7283546567142685, shape=(), dtype=float64)
    loss tf.Tensor(0.6535131927492153, shape=(), dtype=float64) val_loss tf.Tensor(0.7533490931411693, shape=(), dtype=float64)
    loss tf.Tensor(0.8283786906555124, shape=(), dtype=float64) val_loss tf.Tensor(0.7783017180758891, shape=(), dtype=float64)
    loss tf.Tensor(0.7533611493984773, shape=(), dtype=float64) val_loss tf.Tensor(0.6784726110488796, shape=(), dtype=float64)
    loss tf.Tensor(0.704116623233452, shape=(), dtype=float64) val_loss tf.Tensor(0.7784643362784186, shape=(), dtype=float64)
    loss tf.Tensor(0.703390756582842, shape=(), dtype=float64) val_loss tf.Tensor(0.7032723412273862, shape=(), dtype=float64)
    loss tf.Tensor(0.7284076625572362, shape=(), dtype=float64) val_loss tf.Tensor(0.7784269106768464, shape=(), dtype=float64)
    loss tf.Tensor(0.778221016721072, shape=(), dtype=float64) val_loss tf.Tensor(0.8533494611654397, shape=(), dtype=float64)
    loss tf.Tensor(0.7533284917844537, shape=(), dtype=float64) val_loss tf.Tensor(0.7532790832719274, shape=(), dtype=float64)
    loss tf.Tensor(0.8036342303892912, shape=(), dtype=float64) val_loss tf.Tensor(0.7033634238899007, shape=(), dtype=float64)
    loss tf.Tensor(0.7535390795082351, shape=(), dtype=float64) val_loss tf.Tensor(0.7033744109229243, shape=(), dtype=float64)
    loss tf.Tensor(0.6283602511956633, shape=(), dtype=float64) val_loss tf.Tensor(0.8283129716652609, shape=(), dtype=float64)
    loss tf.Tensor(0.7285388992589688, shape=(), dtype=float64) val_loss tf.Tensor(0.7532914829058963, shape=(), dtype=float64)
    loss tf.Tensor(0.6785633737398258, shape=(), dtype=float64) val_loss tf.Tensor(0.753440876000771, shape=(), dtype=float64)
    loss tf.Tensor(0.728778706116581, shape=(), dtype=float64) val_loss tf.Tensor(0.7783316322685636, shape=(), dtype=float64)
    loss tf.Tensor(0.7286722085457048, shape=(), dtype=float64) val_loss tf.Tensor(0.7532500743220181, shape=(), dtype=float64)
    loss tf.Tensor(0.753320398524011, shape=(), dtype=float64) val_loss tf.Tensor(0.7533403383343072, shape=(), dtype=float64)
    loss tf.Tensor(0.8533338582865444, shape=(), dtype=float64) val_loss tf.Tensor(0.7784162706826778, shape=(), dtype=float64)
    loss tf.Tensor(0.7034191225864005, shape=(), dtype=float64) val_loss tf.Tensor(0.7784408751603162, shape=(), dtype=float64)
    loss tf.Tensor(0.7784048084567537, shape=(), dtype=float64) val_loss tf.Tensor(0.7035336624657285, shape=(), dtype=float64)
    loss tf.Tensor(0.7032849380653443, shape=(), dtype=float64) val_loss tf.Tensor(0.7533805142628731, shape=(), dtype=float64)
    loss tf.Tensor(0.8033403917415273, shape=(), dtype=float64) val_loss tf.Tensor(0.7532628579965575, shape=(), dtype=float64)
    loss tf.Tensor(0.778274088483718, shape=(), dtype=float64) val_loss tf.Tensor(0.803258790836421, shape=(), dtype=float64)
    loss tf.Tensor(0.7284473422875088, shape=(), dtype=float64) val_loss tf.Tensor(0.8033537797825481, shape=(), dtype=float64)
    loss tf.Tensor(0.8284304491868875, shape=(), dtype=float64) val_loss tf.Tensor(0.7033828352609082, shape=(), dtype=float64)
    loss tf.Tensor(0.8282936526883351, shape=(), dtype=float64) val_loss tf.Tensor(0.7784444209687258, shape=(), dtype=float64)
    loss tf.Tensor(0.7533774959323084, shape=(), dtype=float64) val_loss tf.Tensor(0.7283087381297554, shape=(), dtype=float64)
    loss tf.Tensor(0.8032914581688239, shape=(), dtype=float64) val_loss tf.Tensor(0.7033245631931291, shape=(), dtype=float64)
    loss tf.Tensor(0.803367037992361, shape=(), dtype=float64) val_loss tf.Tensor(0.7034229251091851, shape=(), dtype=float64)
    loss tf.Tensor(0.703674891322912, shape=(), dtype=float64) val_loss tf.Tensor(0.7034470461711948, shape=(), dtype=float64)
    loss tf.Tensor(0.7532525550179352, shape=(), dtype=float64) val_loss tf.Tensor(0.7033006697212171, shape=(), dtype=float64)
    loss tf.Tensor(0.7536761168345661, shape=(), dtype=float64) val_loss tf.Tensor(0.7532538054208489, shape=(), dtype=float64)
    loss tf.Tensor(0.7536546781899522, shape=(), dtype=float64) val_loss tf.Tensor(0.7033189374061586, shape=(), dtype=float64)
    loss tf.Tensor(0.8037083499016298, shape=(), dtype=float64) val_loss tf.Tensor(0.7034466137543968, shape=(), dtype=float64)
    loss tf.Tensor(0.8044590001714678, shape=(), dtype=float64) val_loss tf.Tensor(0.65332372774462, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
3200 1600 800 400
```

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

```
(0.9282946837413355, 0.5296907809668986)
```

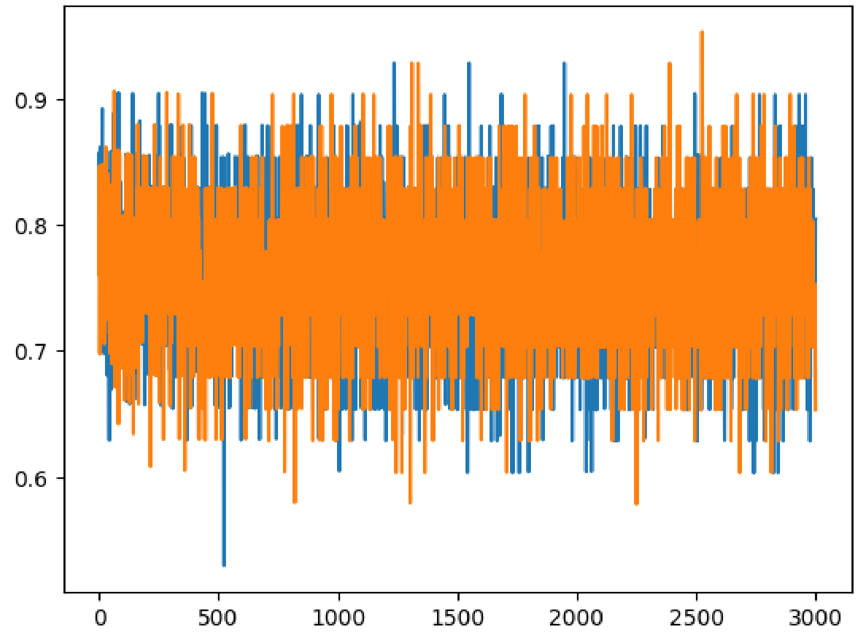
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.9532302407114722, 0.5782771798055311)
```

```
print(w.numpy())
print(b.numpy())
```

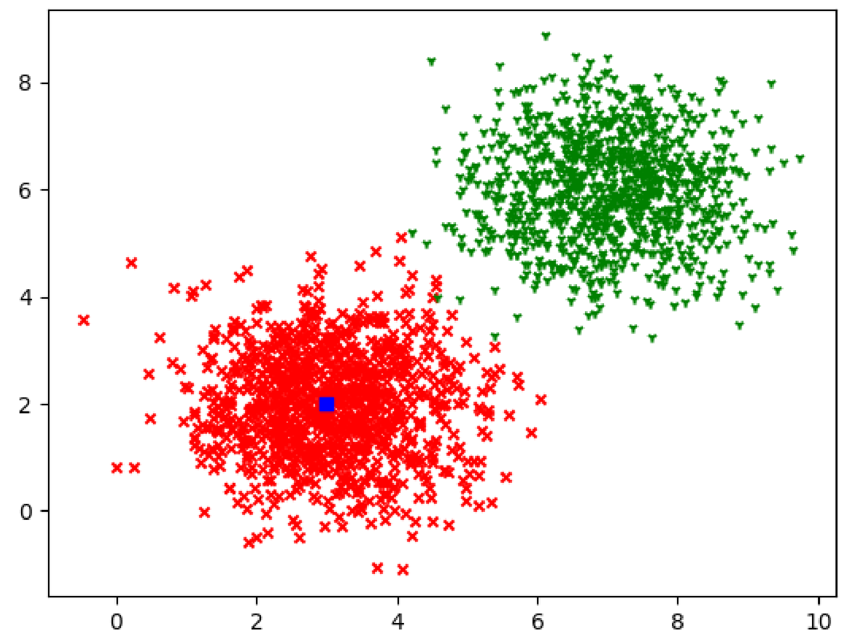
```
[[ 1.42357278 -0.3543947 ]
 [ 1.14629852 -0.13098069]]
[0.6432668  0.10318305]
```

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

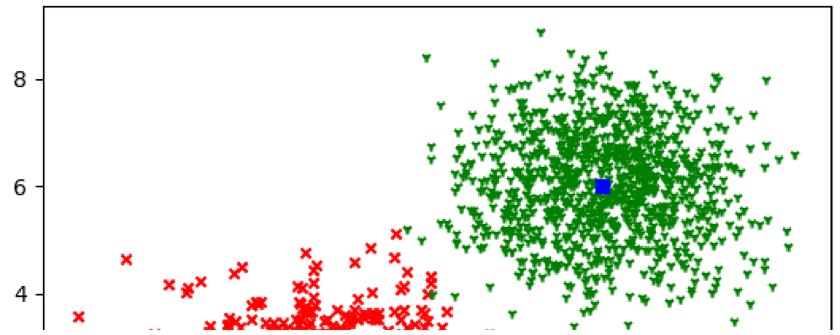


```
x=3.0
```

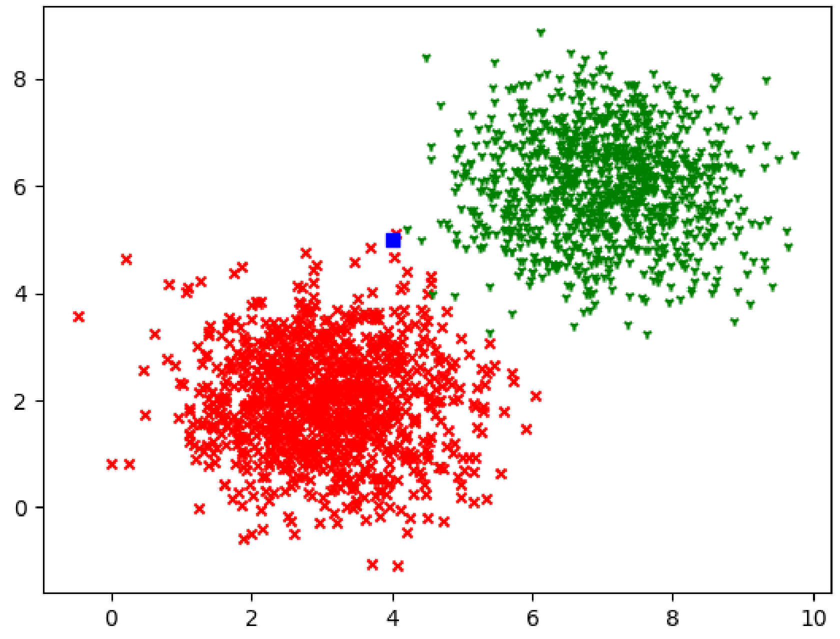
```
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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

Batch size - 10

```
Loss = []
Val_loss = []
epochs = 5000
learning_rate = 0.1
batch_size = 10

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
    loss tf.Tensor(0.514113792926263, shape=(), dtype=float64) val_loss tf.Tensor(0.5100331743740092, shape=(), dtype=float64)
    loss tf.Tensor(0.5096694293137294, shape=(), dtype=float64) val_loss tf.Tensor(0.5181277790226497, shape=(), dtype=float64)
    loss tf.Tensor(0.5154135442723452, shape=(), dtype=float64) val_loss tf.Tensor(0.52041582878872, shape=(), dtype=float64)
    loss tf.Tensor(0.5176992249696943, shape=(), dtype=float64) val_loss tf.Tensor(0.5211597038437501, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
    3200 1600 800 400
```

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

```
    (0.8840225828888627, 0.505522010036187)
```

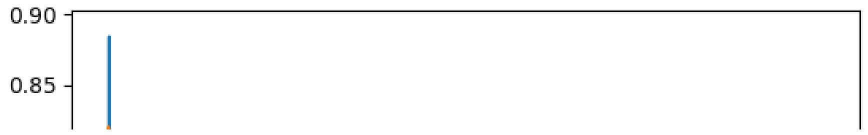
```
np.max(Val_loss),np.min(Val_loss)
```

```
    (0.8200778241943407, 0.5052412980904848)
```

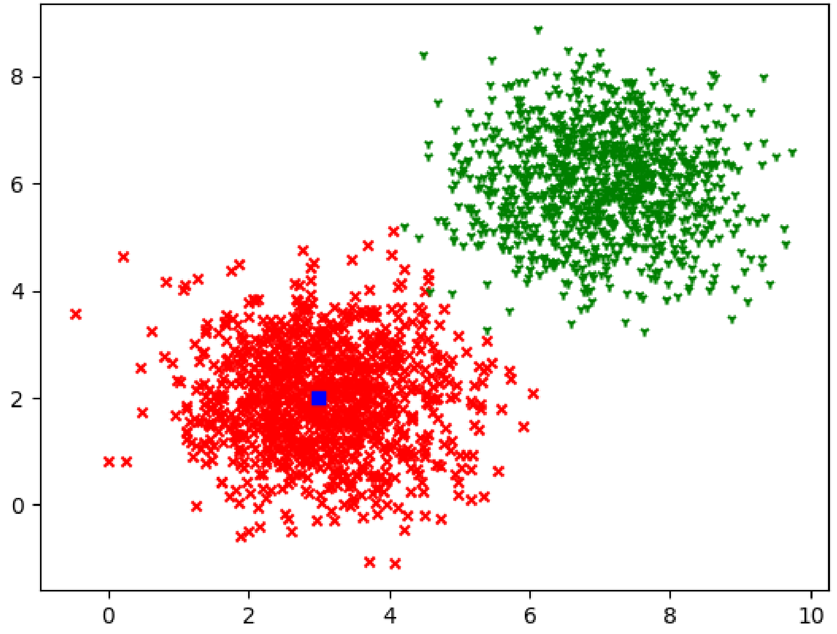
```
print(w.numpy())
print(b.numpy())
```

```
    [[0.22663319 1.15123026]
     [0.20432411 1.49084049]]
    [ 5.1085927  -4.24261619]
```

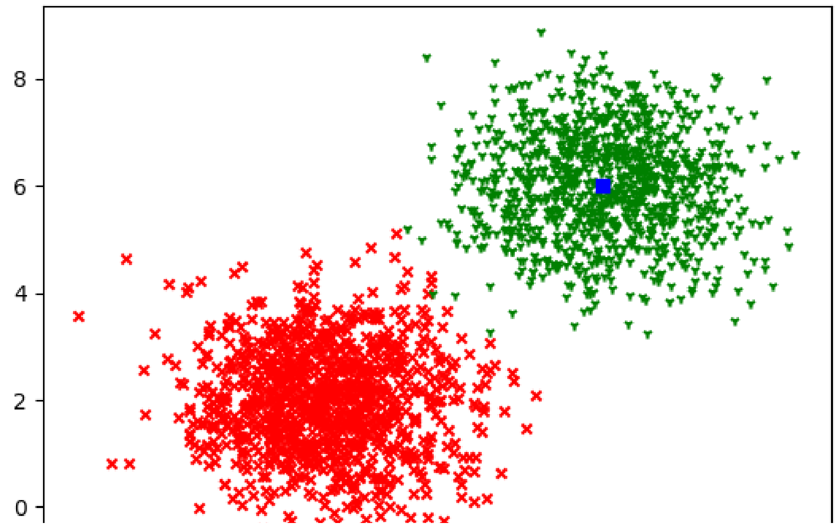
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



```
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 = []
Val_loss = []
epochs = 5000
learning_rate = 0.1
batch_size = 100

w = tf.Variable(np.random.random((2, 2)))
b = tf.Variable(np.random.random((2)))
data = np.column_stack((xs,ys))
data_train,label_train,data_val,label_val = split_dataset(data,labels,0.2)
for _ in range(epochs):

  data_batch,labels_batch = subset_dataset_concatenated(data_train,label_train,batch_size)
  data_val_batch,labels_val_batch = subset_dataset_concatenated(data_val,label_val,batch_size)

  with tf.GradientTape() as tape:

    pred_l=tf.nn.softmax(tf.matmul(data_batch, w) + b)
    pred_l_val=tf.nn.softmax(tf.matmul(data_val_batch, w) + b)

    labels_batch_one_hot = tf.one_hot(labels_batch, depth=2)
    labels_val_batch_one_hot = tf.one_hot(labels_val_batch, depth=2)


    loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_batch_one_hot, pred_l)
    Loss.append(loss.numpy())

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch_one_hot, pred_l_val)
    Val_loss.append(val_loss.numpy())
    print("loss",loss,"val_loss",val_loss)

  dloss_dw,dloss_db = tape.gradient(loss, [w, b])

  w.assign_sub(learning_rate*dloss_dw )
  b.assign_sub(learning_rate*dloss_db )
```

```
     loss tf.Tensor(0.5142177033116411, shape=(), dtype=float64) val_loss tf.Tensor(0.5219609626303264, shape=(), dtype=float64)
     loss tf.Tensor(0.5175655631292828, shape=(), dtype=float64) val_loss tf.Tensor(0.5187147889281255, shape=(), dtype=float64)
     loss tf.Tensor(0.5166561068743636, shape=(), dtype=float64) val_loss tf.Tensor(0.5181992710894627, shape=(), dtype=float64)
```

```
print(data_train.size,label_train.size,data_val.size,label_val.size)
```

```
     3200 1600 800 400
```

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

```
     (0.8007723652234441, 0.5113890253787418)
```

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

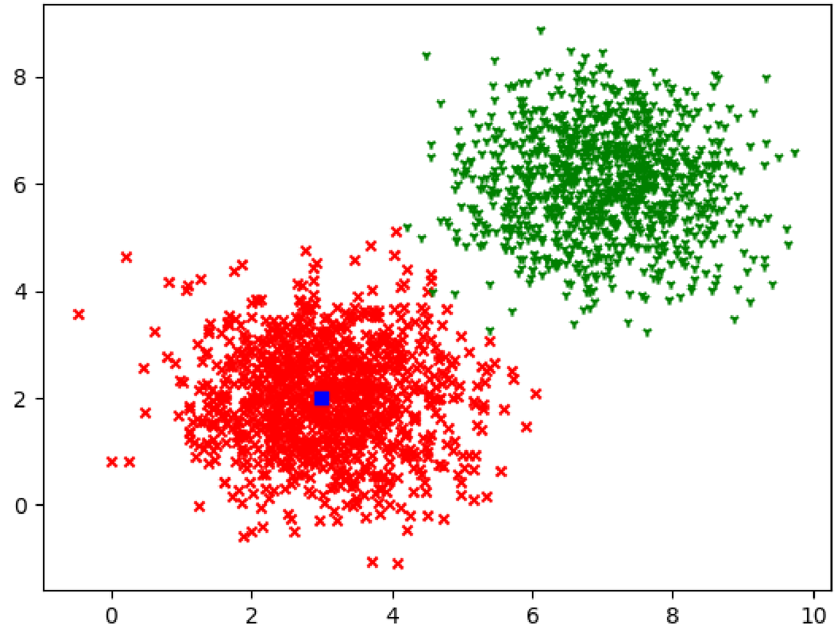
```
     (0.8183081053596879, 0.511993478640346)
```

```
print(w.numpy())
print(b.numpy())
```

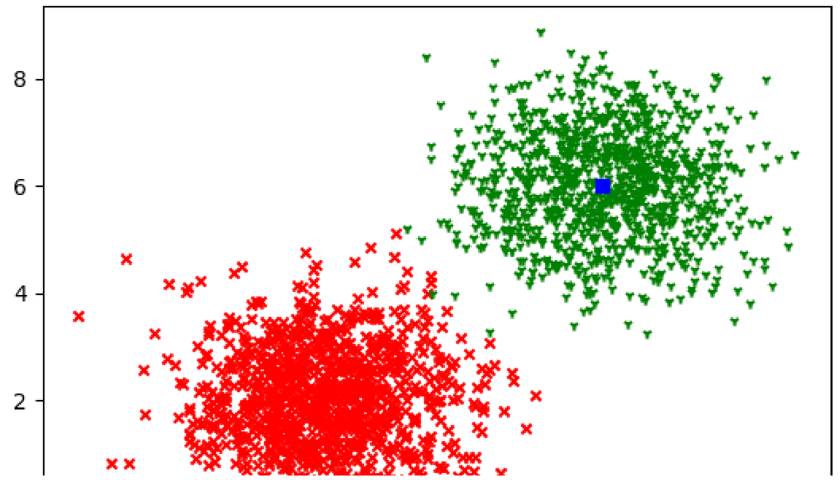
```
     [[ 0.35655663  1.24142172]
      [-0.00202623  1.30862547]]
     [ 5.07338022 -4.22840152]
```

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

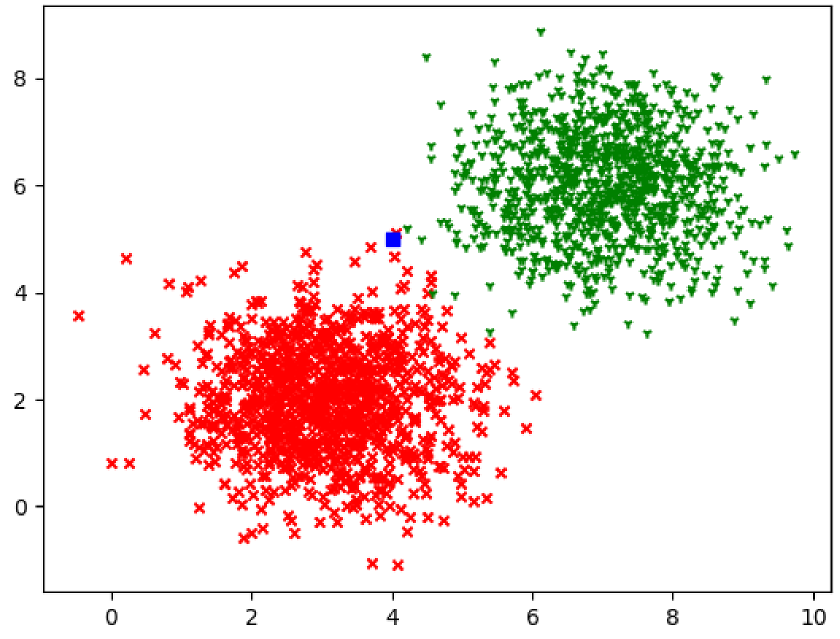
```
x=3.0
y=2.0
plt.scatter(x_label1, y_label1, c='r', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='g', marker='1', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
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()
```



Najlepsze wyniki otrzymałem dla współczynnika uczenia 0.1, liczby epok 5000, batcha równego 20, najgorsze dla współczynnika uczenia 0.001, liczby epok 100, batcha równego 100.

I got the best results for a learning rate of 0.1, a number of epochs of 5000, a batch of 20, and the worst for a learning rate of 0.001, a number of