Exercise 5

```
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
```

Three gangs

Softmax regresion

```
s = [0.2, 0.1, 0.6, 0.1]
exps = [np.exp(i) for i in s]
sum_of_exps = sum(exps)
softmax = [j/sum_of_exps for j in exps]
```

Dataset:

```
x_label0 = np.random.normal(1, 1.5, (1000, 1))
y_label0 = np.random.normal(1, 1.5, (1000, 1))
x_label1 = np.random.normal(5, 1.5, (1000, 1))
y_label1 = np.random.normal(4, 1.5, (1000, 1))
x_label2 = np.random.normal(8, 1.5, (1000, 1))
y_label2 = np.random.normal(0, 1.5, (1000, 1))

data_label0 = np.concatenate([x_label0, y_label0],axis=1)
data_label1 = np.concatenate([x_label1, y_label1],axis=1)
data_label2 = np.concatenate([x_label2, y_label2],axis=1)
points = np.concatenate([data_label0, data_label1, data_label2],axis=0)
```

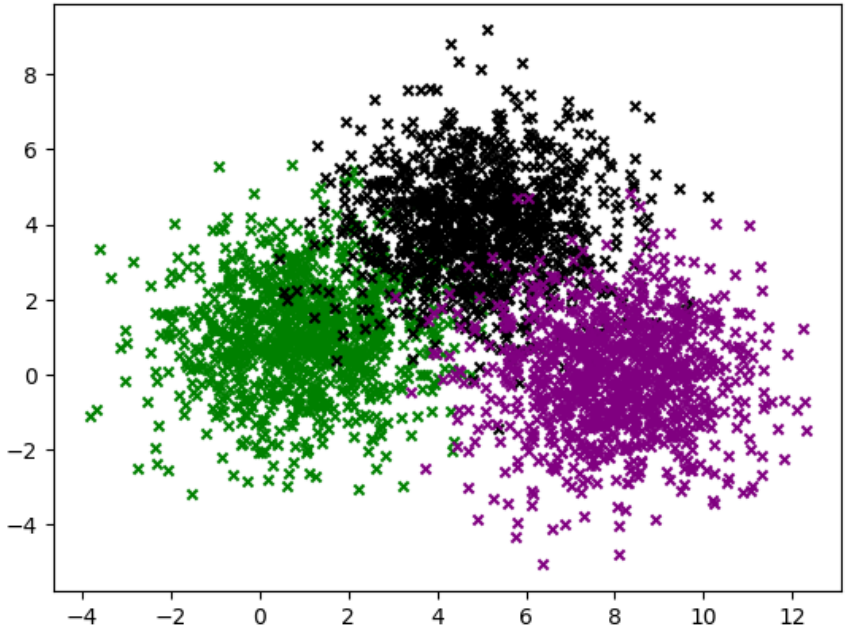
Kodowanie one-hot

```
labels = np.array([[1., 0., 0.]] * len(data_label0) + [[0., 1., 0.]] * len(data_label1) + [[0.,0., 1.]] * len(data_label2))
```

```
points.shape,labels.shape
```

```
((3000, 2), (3000, 3))
```

```
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.show()
```




```
x_label1
```

```
[ 5.61473768],
[ 5.71940417],
[ 4.89756079]])
```

```
#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 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
```

```
(3000, 3)
```

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

w = tf.Variable(np.random.random((2, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)


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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.5865369150521869, shape=(), dtype=float64) val_loss tf.Tensor(0.6069459186691029, shape=(), dtype=float64)
loss tf.Tensor(0.5955029696727389, shape=(), dtype=float64) val_loss tf.Tensor(0.5880171514786563, shape=(), dtype=float64)
loss tf.Tensor(0.5871016356596991, shape=(), dtype=float64) val_loss tf.Tensor(0.5958575941020829, shape=(), dtype=float64)
loss tf.Tensor(0.6091248101044586, shape=(), dtype=float64) val_loss tf.Tensor(0.5938308481774592, shape=(), dtype=float64)
loss tf.Tensor(0.6307364884919273, shape=(), dtype=float64) val_loss tf.Tensor(0.6007758995997488, shape=(), dtype=float64)
loss tf.Tensor(0.609325952788007, shape=(), dtype=float64) val_loss tf.Tensor(0.5874555788527258, shape=(), dtype=float64)
loss tf.Tensor(0.6016983747682965, shape=(), dtype=float64) val_loss tf.Tensor(0.6147635144439626, shape=(), dtype=float64)
loss tf.Tensor(0.6072281581262058, shape=(), dtype=float64) val_loss tf.Tensor(0.5885336795788765, shape=(), dtype=float64)
loss tf.Tensor(0.6081162539876972, shape=(), dtype=float64) val_loss tf.Tensor(0.5975983565844272, shape=(), dtype=float64)
loss tf.Tensor(0.6190551866800501, shape=(), dtype=float64) val_loss tf.Tensor(0.5831119667947935, shape=(), dtype=float64)
loss tf.Tensor(0.6160555972206369, shape=(), dtype=float64) val_loss tf.Tensor(0.6086720997413795, shape=(), dtype=float64)
loss tf.Tensor(0.5906805797116782, shape=(), dtype=float64) val_loss tf.Tensor(0.5930969313785477, shape=(), dtype=float64)
loss tf.Tensor(0.6082623428906968, shape=(), dtype=float64) val_loss tf.Tensor(0.5962142181713302, shape=(), dtype=float64)
loss tf.Tensor(0.60772999945103, shape=(), dtype=float64) val_loss tf.Tensor(0.5954432812105773, shape=(), dtype=float64)
loss tf.Tensor(0.6130175015074907, shape=(), dtype=float64) val_loss tf.Tensor(0.5938514216018614, shape=(), dtype=float64)
loss tf.Tensor(0.5740197208486724, shape=(), dtype=float64) val_loss tf.Tensor(0.5975533982061615, shape=(), dtype=float64)
loss tf.Tensor(0.6023007825549389, shape=(), dtype=float64) val_loss tf.Tensor(0.6234731157231262, shape=(), dtype=float64)
loss tf.Tensor(0.5968324665984375, shape=(), dtype=float64) val_loss tf.Tensor(0.6336800069792188, shape=(), dtype=float64)
loss tf.Tensor(0.5973964626307888, shape=(), dtype=float64) val_loss tf.Tensor(0.5920077393178553, shape=(), dtype=float64)
loss tf.Tensor(0.6181542024237532, shape=(), dtype=float64) val_loss tf.Tensor(0.5879281886080296, shape=(), dtype=float64)
loss tf.Tensor(0.6074763080765256, shape=(), dtype=float64) val_loss tf.Tensor(0.6211771045005776, shape=(), dtype=float64)
loss tf.Tensor(0.5728384434765148, shape=(), dtype=float64) val_loss tf.Tensor(0.6225582002773804, shape=(), dtype=float64)
loss tf.Tensor(0.5885131329502188, shape=(), dtype=float64) val_loss tf.Tensor(0.5999205225631936, shape=(), dtype=float64)
loss tf.Tensor(0.6131805580038426, shape=(), dtype=float64) val_loss tf.Tensor(0.5961401557672052, shape=(), dtype=float64)
loss tf.Tensor(0.590868415147505, shape=(), dtype=float64) val_loss tf.Tensor(0.6495854344307046, shape=(), dtype=float64)
loss tf.Tensor(0.6089592670555997, shape=(), dtype=float64) val_loss tf.Tensor(0.6110738181787603, shape=(), dtype=float64)
loss tf.Tensor(0.5962446797679704, shape=(), dtype=float64) val_loss tf.Tensor(0.5897596892185856, shape=(), dtype=float64)
loss tf.Tensor(0.6196477703919365, shape=(), dtype=float64) val_loss tf.Tensor(0.5933902376852718, shape=(), dtype=float64)
loss tf.Tensor(0.5904116799142543, shape=(), dtype=float64) val_loss tf.Tensor(0.6124514437538061, shape=(), dtype=float64)
loss tf.Tensor(0.6286410332319727, shape=(), dtype=float64) val_loss tf.Tensor(0.5967008276821872, shape=(), dtype=float64)
loss tf.Tensor(0.5819331385516378, shape=(), dtype=float64) val_loss tf.Tensor(0.5833038797958107, shape=(), dtype=float64)
loss tf.Tensor(0.5935299701252547, shape=(), dtype=float64) val_loss tf.Tensor(0.5960152088329467, shape=(), dtype=float64)
loss tf.Tensor(0.593993732622453, shape=(), dtype=float64) val_loss tf.Tensor(0.5868224228863121, shape=(), dtype=float64)
loss tf.Tensor(0.6116887646978572, shape=(), dtype=float64) val_loss tf.Tensor(0.6140872196869906, shape=(), dtype=float64)
loss tf.Tensor(0.6052627422924404, shape=(), dtype=float64) val_loss tf.Tensor(0.6056131192047761, shape=(), dtype=float64)
loss tf.Tensor(0.5783517483923992, shape=(), dtype=float64) val_loss tf.Tensor(0.5887944535162807, shape=(), dtype=float64)
loss tf.Tensor(0.6108306773033834, shape=(), dtype=float64) val_loss tf.Tensor(0.5932721480885568, shape=(), dtype=float64)
loss tf.Tensor(0.5962680401805158, shape=(), dtype=float64) val_loss tf.Tensor(0.5898454901376284, shape=(), dtype=float64)
loss tf.Tensor(0.6234404969573258, shape=(), dtype=float64) val_loss tf.Tensor(0.5966940672416807, shape=(), dtype=float64)
loss tf.Tensor(0.61946573531729, shape=(), dtype=float64) val_loss tf.Tensor(0.6133021829861427, shape=(), dtype=float64)
loss tf.Tensor(0.5915980008558912, shape=(), dtype=float64) val_loss tf.Tensor(0.5905239421091937, shape=(), dtype=float64)
loss tf.Tensor(0.5971479020257776, shape=(), dtype=float64) val_loss tf.Tensor(0.6058348443242758, shape=(), dtype=float64)
loss tf.Tensor(0.6009797860915965, shape=(), dtype=float64) val_loss tf.Tensor(0.6028099090970211, shape=(), dtype=float64)
loss tf.Tensor(0.5831340385175734, shape=(), dtype=float64) val_loss tf.Tensor(0.5914461446414203, shape=(), dtype=float64)
loss tf.Tensor(0.5981991454358305, shape=(), dtype=float64) val_loss tf.Tensor(0.6002424356825657, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8227390251926595, 0.572023823731623)
```

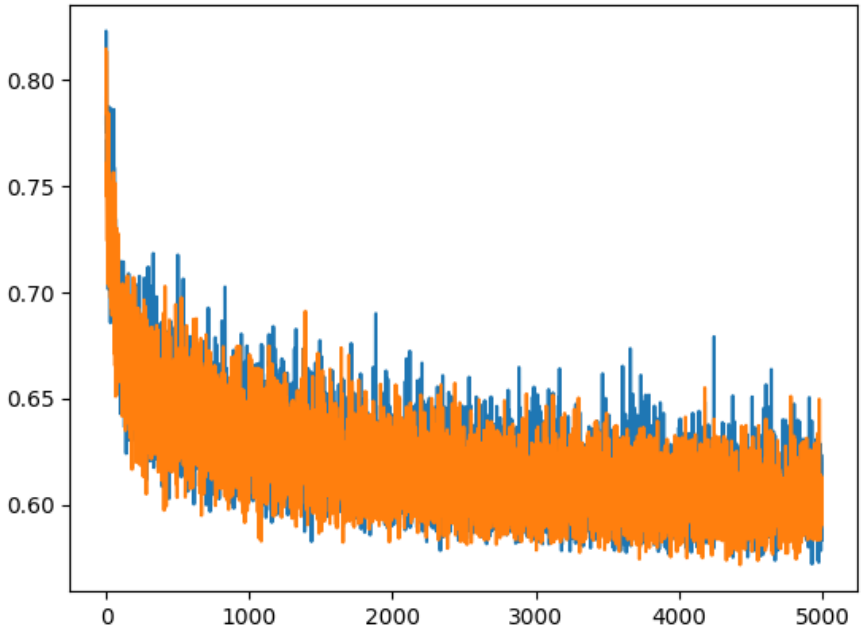
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8142714468370269, 0.5716919853930085)
```

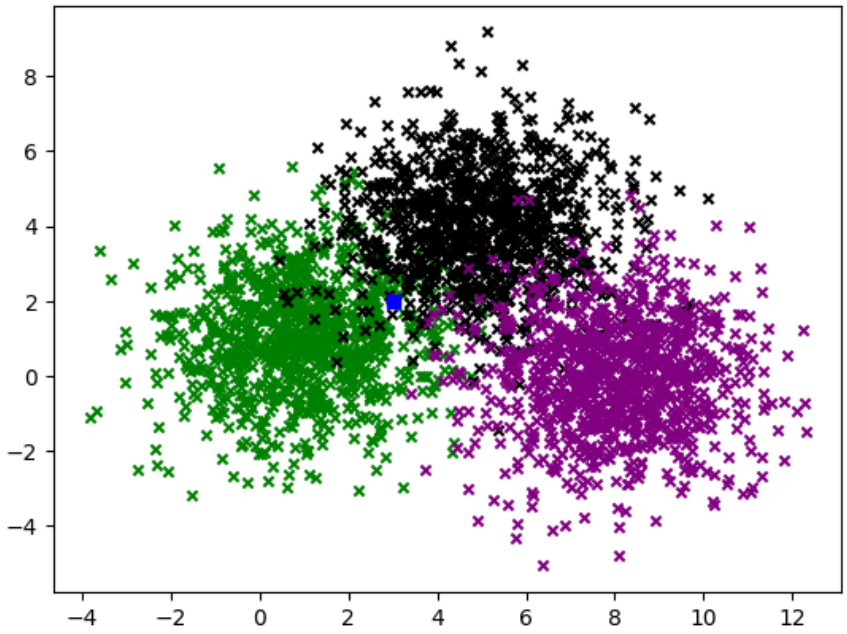
```
print(w.numpy())
print(b.numpy())
```

```
[[-0.77754904  0.20680772  0.97694973]
 [ 0.51297229  1.7580681  -0.45759847]]
[ 4.57287348 -1.35844996 -1.4956957 ]
```

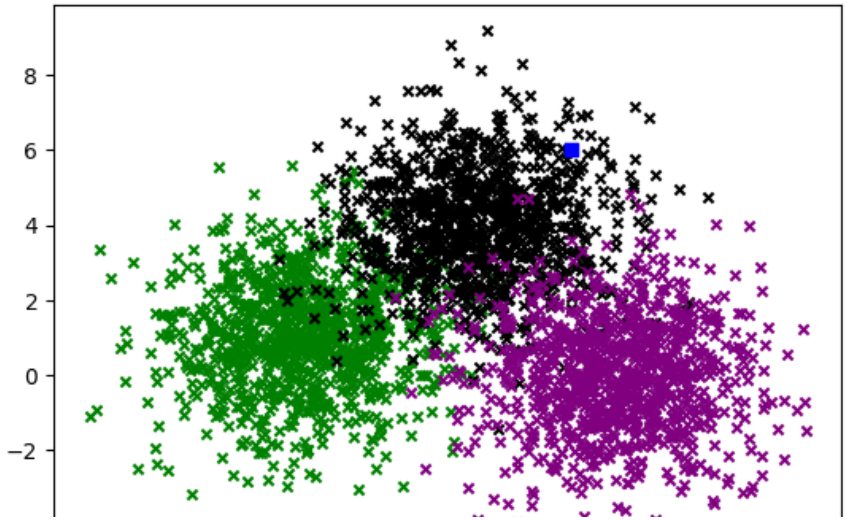
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```

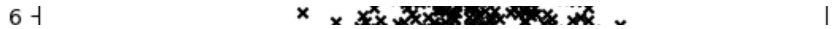



```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)


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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.6771050667421158, shape=(), dtype=float64) val_loss tf.Tensor(0.6252687885651552, shape=(), dtype=float64)
loss tf.Tensor(0.6503702209252161, shape=(), dtype=float64) val_loss tf.Tensor(0.6467836034024195, shape=(), dtype=float64)
loss tf.Tensor(0.657736106732645, shape=(), dtype=float64) val_loss tf.Tensor(0.6749055253783007, shape=(), dtype=float64)
loss tf.Tensor(0.6774174654454237, shape=(), dtype=float64) val_loss tf.Tensor(0.6638429760988147, shape=(), dtype=float64)
loss tf.Tensor(0.6291552086142538, shape=(), dtype=float64) val_loss tf.Tensor(0.6235274397249277, shape=(), dtype=float64)
loss tf.Tensor(0.6769521657859358, shape=(), dtype=float64) val_loss tf.Tensor(0.6350681563218535, shape=(), dtype=float64)
loss tf.Tensor(0.6558181007327741, shape=(), dtype=float64) val_loss tf.Tensor(0.6557583729803833, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8046920774927587, 0.5986084332051266)
```

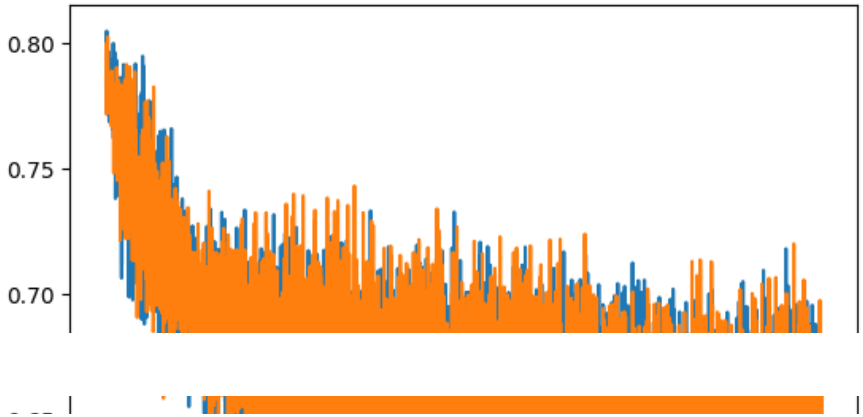
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8024314901495992, 0.5923339050351574)
```

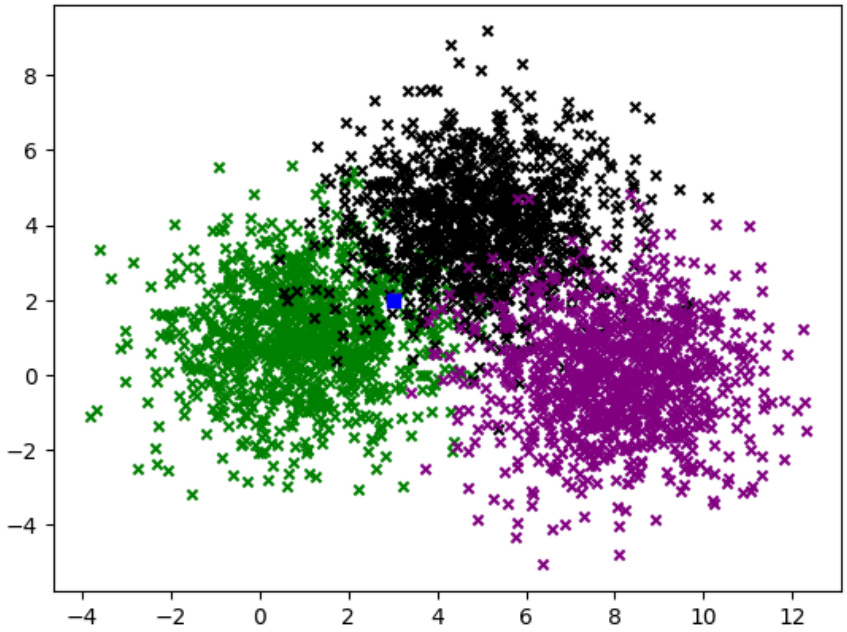
```
print(w.numpy())
print(b.numpy())
```

```
[[ 0.15632094  0.75954839  1.24254765]
 [ 1.15656334  1.36019651 -0.08130425]]
[ 0.89811175 -0.08275817 -0.18440522]
```

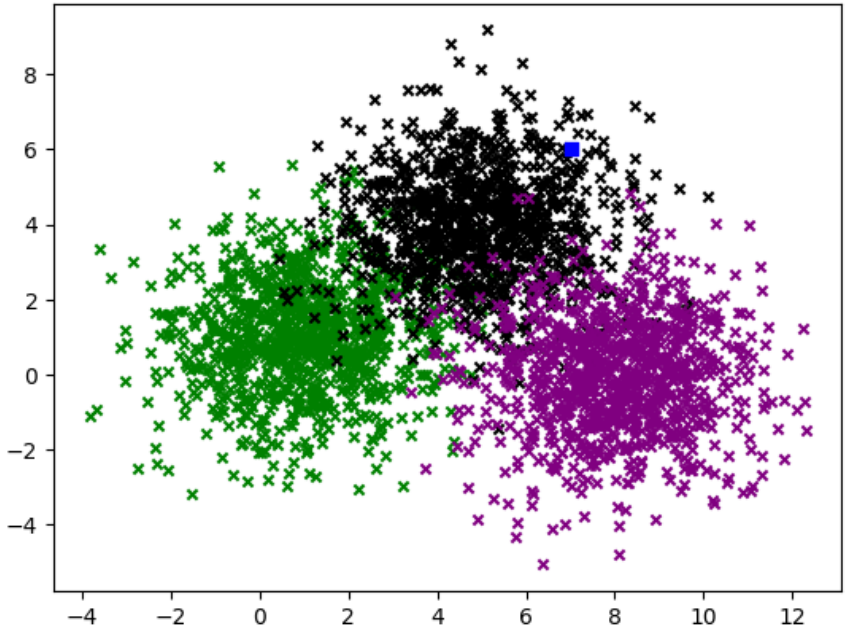
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



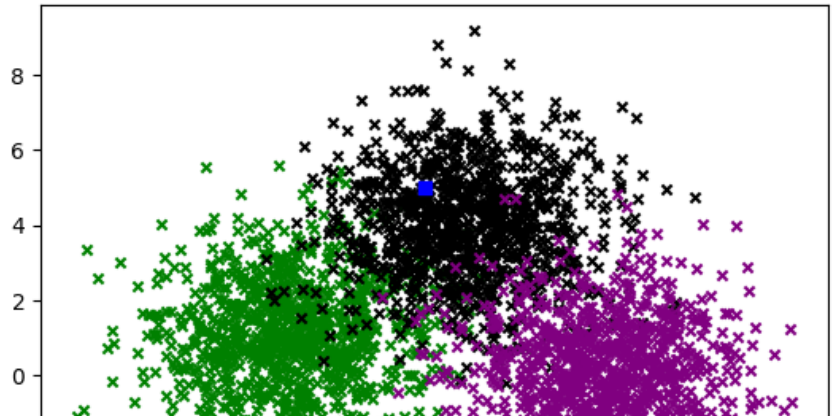

```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



Learning rate 0.001

```
Loss = []
Val_loss = []
epochs = 5000
learning_rate = 0.001
batch_size = 20

w = tf.Variable(np.random.random((2, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)


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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.8678682559898366, shape=(), dtype=float64) val_loss tf.Tensor(0.8155469785858489, shape=(), dtype=float64)
loss tf.Tensor(0.8131320432694178, shape=(), dtype=float64) val_loss tf.Tensor(0.8447403063677191, shape=(), dtype=float64)
loss tf.Tensor(0.8201491084873819, shape=(), dtype=float64) val_loss tf.Tensor(0.841090501811164, shape=(), dtype=float64)
loss tf.Tensor(0.8045614283977031, shape=(), dtype=float64) val_loss tf.Tensor(0.7981163783575157, shape=(), dtype=float64)
loss tf.Tensor(0.8085359746596719, shape=(), dtype=float64) val_loss tf.Tensor(0.8288862731278683, shape=(), dtype=float64)
loss tf.Tensor(0.8231829086155542, shape=(), dtype=float64) val_loss tf.Tensor(0.7985759396034033, shape=(), dtype=float64)
loss tf.Tensor(0.791419748964286, shape=(), dtype=float64) val_loss tf.Tensor(0.8224690983763505, shape=(), dtype=float64)
loss tf.Tensor(0.7484066284394925, shape=(), dtype=float64) val_loss tf.Tensor(0.7844213982527941, shape=(), dtype=float64)
loss tf.Tensor(0.8027850932452185, shape=(), dtype=float64) val_loss tf.Tensor(0.8264628715086042, shape=(), dtype=float64)
loss tf.Tensor(0.8077156568038693, shape=(), dtype=float64) val_loss tf.Tensor(0.8295982966965324, shape=(), dtype=float64)
loss tf.Tensor(0.7812469713403265, shape=(), dtype=float64) val_loss tf.Tensor(0.8173661626431056, shape=(), dtype=float64)
loss tf.Tensor(0.7939597472033098, shape=(), dtype=float64) val_loss tf.Tensor(0.8167015981544313, shape=(), dtype=float64)
loss tf.Tensor(0.7903005242328651, shape=(), dtype=float64) val_loss tf.Tensor(0.8071068213009243, shape=(), dtype=float64)
loss tf.Tensor(0.8452960892303911, shape=(), dtype=float64) val_loss tf.Tensor(0.8180572558840009, shape=(), dtype=float64)
loss tf.Tensor(0.815176337812224, shape=(), dtype=float64) val_loss tf.Tensor(0.8362219608346809, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8925800397751317, 0.7366488493781577)
```

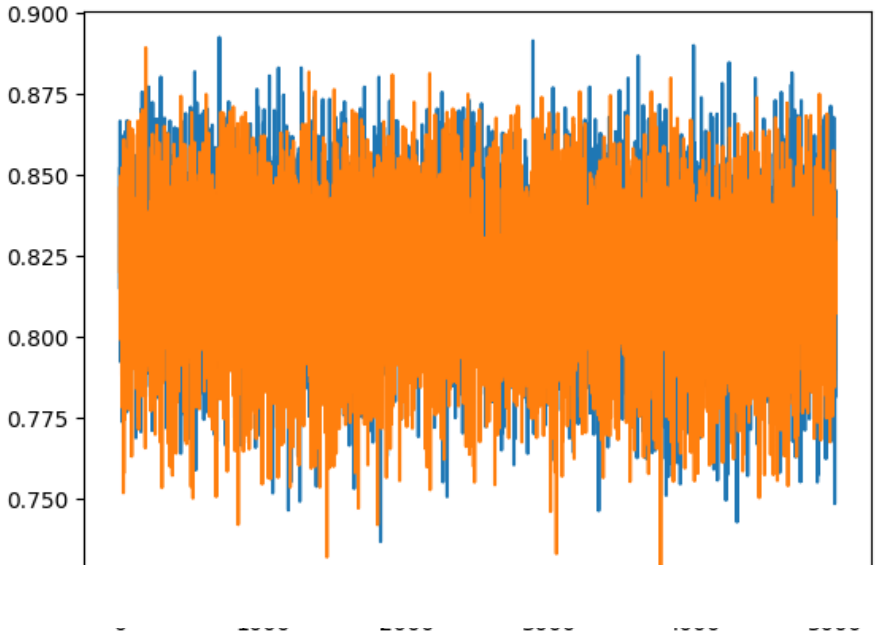
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8893945746369688, 0.7276200280485948)
```

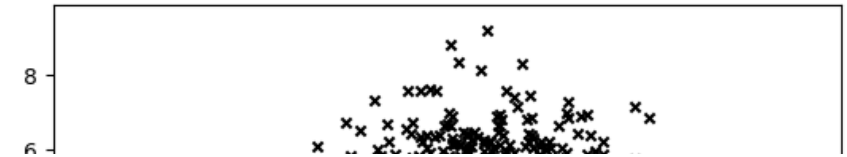
```
print(w.numpy())
print(b.numpy())
```

```
[[0.71594753 0.22610428 0.27433853]
 [0.84671485 0.5327345  0.49550385]]
[1.0365432  0.85105495 0.56822219]
```

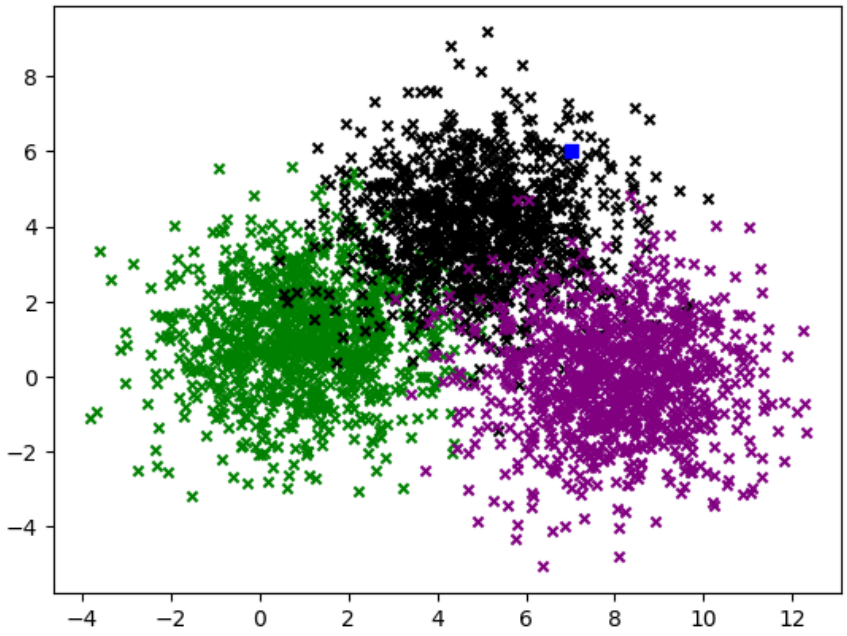
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



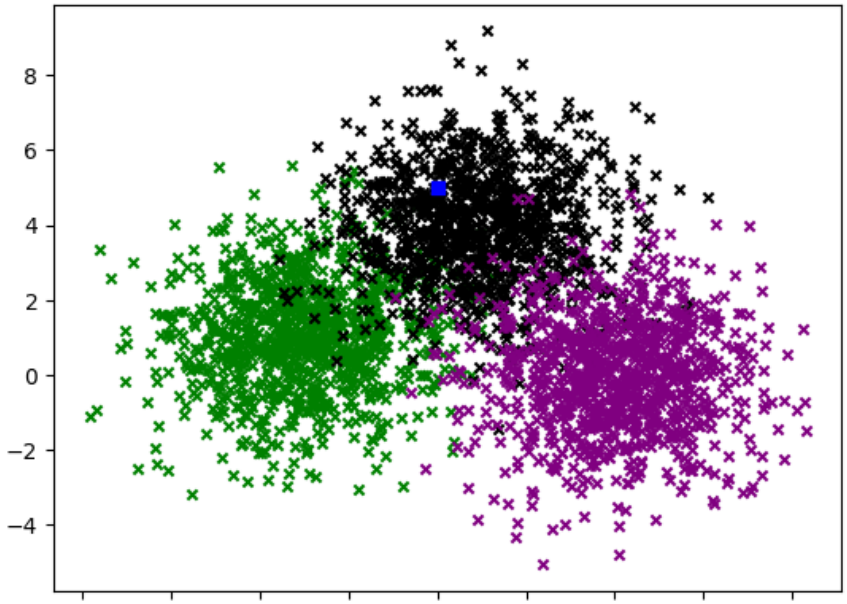

```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)
```

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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.7128258605627058, shape=(), dtype=float64) val_loss tf.Tensor(0.7228547645793836, shape=(), dtype=float64)
loss tf.Tensor(0.7817485362989076, shape=(), dtype=float64) val_loss tf.Tensor(0.7162157953809023, shape=(), dtype=float64)
loss tf.Tensor(0.7779540569550842, shape=(), dtype=float64) val_loss tf.Tensor(0.7469176216967078, shape=(), dtype=float64)
loss tf.Tensor(0.7904320406807612, shape=(), dtype=float64) val_loss tf.Tensor(0.7828260668366201, shape=(), dtype=float64)
loss tf.Tensor(0.7280769058762917, shape=(), dtype=float64) val_loss tf.Tensor(0.7604257113437757, shape=(), dtype=float64)
loss tf.Tensor(0.7582216501459058, shape=(), dtype=float64) val_loss tf.Tensor(0.7823119792676132, shape=(), dtype=float64)
loss tf.Tensor(0.7427707641338037, shape=(), dtype=float64) val_loss tf.Tensor(0.7759798971156021, shape=(), dtype=float64)
loss tf.Tensor(0.7254754031603087, shape=(), dtype=float64) val_loss tf.Tensor(0.7615664857673112, shape=(), dtype=float64)
loss tf.Tensor(0.7775848083388549, shape=(), dtype=float64) val_loss tf.Tensor(0.7821800531334423, shape=(), dtype=float64)
loss tf.Tensor(0.7362172258137256, shape=(), dtype=float64) val_loss tf.Tensor(0.7358282306540282, shape=(), dtype=float64)
loss tf.Tensor(0.7587812830318924, shape=(), dtype=float64) val_loss tf.Tensor(0.7275177002997815, shape=(), dtype=float64)
loss tf.Tensor(0.7775535399913436, shape=(), dtype=float64) val_loss tf.Tensor(0.728120068367535, shape=(), dtype=float64)
loss tf.Tensor(0.7439580020425065, shape=(), dtype=float64) val_loss tf.Tensor(0.6859204223397826, shape=(), dtype=float64)
loss tf.Tensor(0.7308159775329214, shape=(), dtype=float64) val_loss tf.Tensor(0.7601765764954443, shape=(), dtype=float64)
loss tf.Tensor(0.7491248872623772, shape=(), dtype=float64) val_loss tf.Tensor(0.7513057184624061, shape=(), dtype=float64)
loss tf.Tensor(0.7239980160979119, shape=(), dtype=float64) val_loss tf.Tensor(0.7699547545988489, shape=(), dtype=float64)
loss tf.Tensor(0.7562169785129972, shape=(), dtype=float64) val_loss tf.Tensor(0.7767818460561877, shape=(), dtype=float64)
loss tf.Tensor(0.7447373135815645, shape=(), dtype=float64) val_loss tf.Tensor(0.7840546196785599, shape=(), dtype=float64)
loss tf.Tensor(0.6954371106962565, shape=(), dtype=float64) val_loss tf.Tensor(0.7377264530486116, shape=(), dtype=float64)
loss tf.Tensor(0.8011638585755854, shape=(), dtype=float64) val_loss tf.Tensor(0.7309964669118882, shape=(), dtype=float64)
loss tf.Tensor(0.7317739666091345, shape=(), dtype=float64) val_loss tf.Tensor(0.7715745204957609, shape=(), dtype=float64)
loss tf.Tensor(0.7245125636724957, shape=(), dtype=float64) val_loss tf.Tensor(0.7753731418908907, shape=(), dtype=float64)
loss tf.Tensor(0.7879458848164143, shape=(), dtype=float64) val_loss tf.Tensor(0.7390198394655962, shape=(), dtype=float64)
loss tf.Tensor(0.7606098897309702, shape=(), dtype=float64) val_loss tf.Tensor(0.7495177276682006, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8197414709609113, 0.6914993958455458)
```

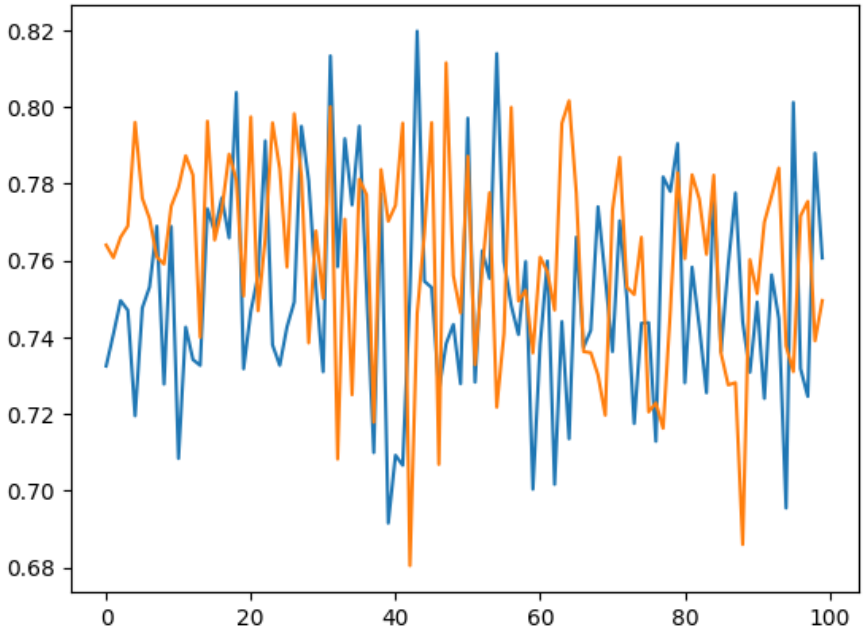
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8114715354201604, 0.6804288885846151)
```

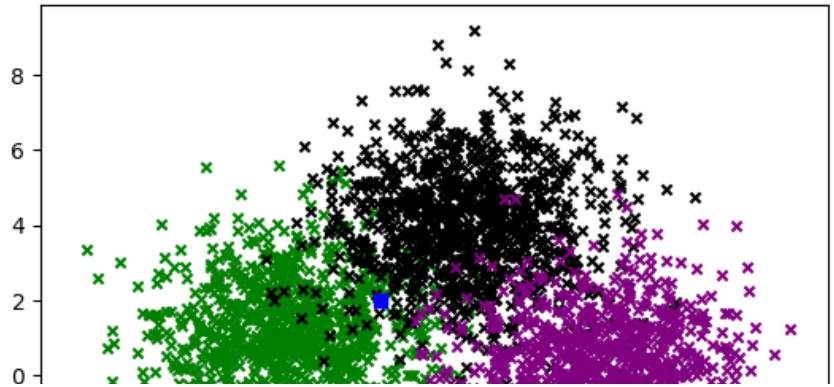
```
print(w.numpy())
print(b.numpy())
```

```
[[0.28031447 0.53164352 0.2734538 ]
 [0.4380458  0.76366019 0.35611346]]
[0.62302722 0.26674981 0.26600703]
```

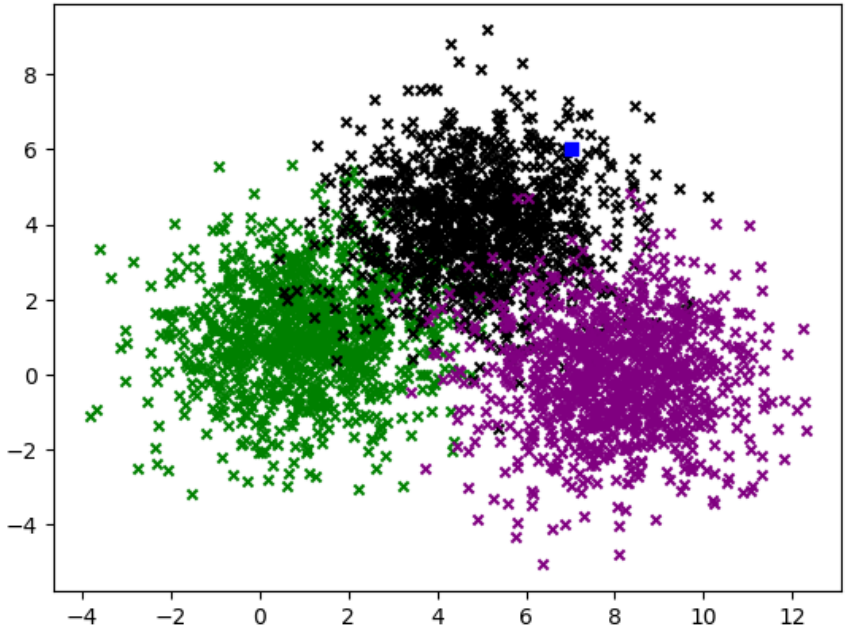
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



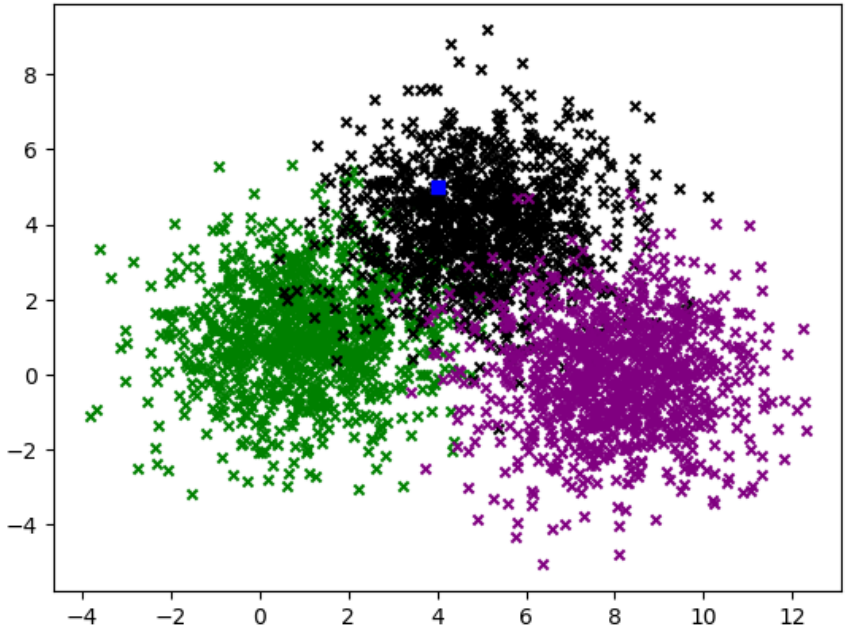

```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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, 3)))
b = tf.Variable(np.random.random((3)))
data = points
```

```
  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)


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

      val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.6058508213723526, shape=(), dtype=float64) val_loss tf.Tensor(0.6156105427056865, shape=(), dtype=float64)
loss tf.Tensor(0.6088813142497618, shape=(), dtype=float64) val_loss tf.Tensor(0.6252284577346289, shape=(), dtype=float64)
loss tf.Tensor(0.6226886850018947, shape=(), dtype=float64) val_loss tf.Tensor(0.5945387475982018, shape=(), dtype=float64)
loss tf.Tensor(0.6353875476936807, shape=(), dtype=float64) val_loss tf.Tensor(0.6419725136097079, shape=(), dtype=float64)
loss tf.Tensor(0.6161693615838393, shape=(), dtype=float64) val_loss tf.Tensor(0.6119881671835863, shape=(), dtype=float64)
loss tf.Tensor(0.6316172229453139, shape=(), dtype=float64) val_loss tf.Tensor(0.6083663917920537, shape=(), dtype=float64)
loss tf.Tensor(0.6426275713162655, shape=(), dtype=float64) val_loss tf.Tensor(0.615255601168632, shape=(), dtype=float64)
loss tf.Tensor(0.6135449003609252, shape=(), dtype=float64) val_loss tf.Tensor(0.656024248643581, shape=(), dtype=float64)
loss tf.Tensor(0.6258340213781052, shape=(), dtype=float64) val_loss tf.Tensor(0.6322420195564209, shape=(), dtype=float64)
loss tf.Tensor(0.6267992310408474, shape=(), dtype=float64) val_loss tf.Tensor(0.6365539970929408, shape=(), dtype=float64)
loss tf.Tensor(0.630471268505623, shape=(), dtype=float64) val_loss tf.Tensor(0.6145969180582963, shape=(), dtype=float64)
loss tf.Tensor(0.5994044111827617, shape=(), dtype=float64) val_loss tf.Tensor(0.6072129680040603, shape=(), dtype=float64)
loss tf.Tensor(0.6120380395845655, shape=(), dtype=float64) val_loss tf.Tensor(0.6105065864968108, shape=(), dtype=float64)
loss tf.Tensor(0.6356493387840164, shape=(), dtype=float64) val_loss tf.Tensor(0.602266209127224, shape=(), dtype=float64)
loss tf.Tensor(0.5998277971501682, shape=(), dtype=float64) val_loss tf.Tensor(0.6184439923740209, shape=(), dtype=float64)
loss tf.Tensor(0.6052457462133167, shape=(), dtype=float64) val_loss tf.Tensor(0.6092681557492214, shape=(), dtype=float64)
loss tf.Tensor(0.5963316334015133, shape=(), dtype=float64) val_loss tf.Tensor(0.6010649867947429, shape=(), dtype=float64)
loss tf.Tensor(0.6285054502614885, shape=(), dtype=float64) val_loss tf.Tensor(0.6122696352840403, shape=(), dtype=float64)
loss tf.Tensor(0.5909199351699168, shape=(), dtype=float64) val_loss tf.Tensor(0.6011895393156913, shape=(), dtype=float64)
loss tf.Tensor(0.5895515205651503, shape=(), dtype=float64) val_loss tf.Tensor(0.6229548612961139, shape=(), dtype=float64)
loss tf.Tensor(0.6200943913967107, shape=(), dtype=float64) val_loss tf.Tensor(0.6149656581836876, shape=(), dtype=float64)
loss tf.Tensor(0.6104544796253809, shape=(), dtype=float64) val_loss tf.Tensor(0.6143261814639164, shape=(), dtype=float64)
loss tf.Tensor(0.6194237492302408, shape=(), dtype=float64) val_loss tf.Tensor(0.6099729447523708, shape=(), dtype=float64)
loss tf.Tensor(0.6300766512525067, shape=(), dtype=float64) val_loss tf.Tensor(0.6079103762928441, shape=(), dtype=float64)
loss tf.Tensor(0.5898680705974952, shape=(), dtype=float64) val_loss tf.Tensor(0.5894700791338876, shape=(), dtype=float64)
loss tf.Tensor(0.6082281364788782, shape=(), dtype=float64) val_loss tf.Tensor(0.6316973294378914, shape=(), dtype=float64)
loss tf.Tensor(0.5880260827272059, shape=(), dtype=float64) val_loss tf.Tensor(0.602584480767371, shape=(), dtype=float64)
loss tf.Tensor(0.6171589566680449, shape=(), dtype=float64) val_loss tf.Tensor(0.6121595416693133, shape=(), dtype=float64)
loss tf.Tensor(0.6117998196341464, shape=(), dtype=float64) val_loss tf.Tensor(0.623894551888064, shape=(), dtype=float64)
loss tf.Tensor(0.596095720633784, shape=(), dtype=float64) val_loss tf.Tensor(0.6193058595996651, shape=(), dtype=float64)
loss tf.Tensor(0.6002164834437915, shape=(), dtype=float64) val_loss tf.Tensor(0.6179486872944767, shape=(), dtype=float64)
loss tf.Tensor(0.6116567853070739, shape=(), dtype=float64) val_loss tf.Tensor(0.5940010244589464, shape=(), dtype=float64)
loss tf.Tensor(0.6452244881991233, shape=(), dtype=float64) val_loss tf.Tensor(0.609819300252243, shape=(), dtype=float64)
loss tf.Tensor(0.5996721745186097, shape=(), dtype=float64) val_loss tf.Tensor(0.6260606559979002, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8231510414371108, 0.5856957068522768)
```

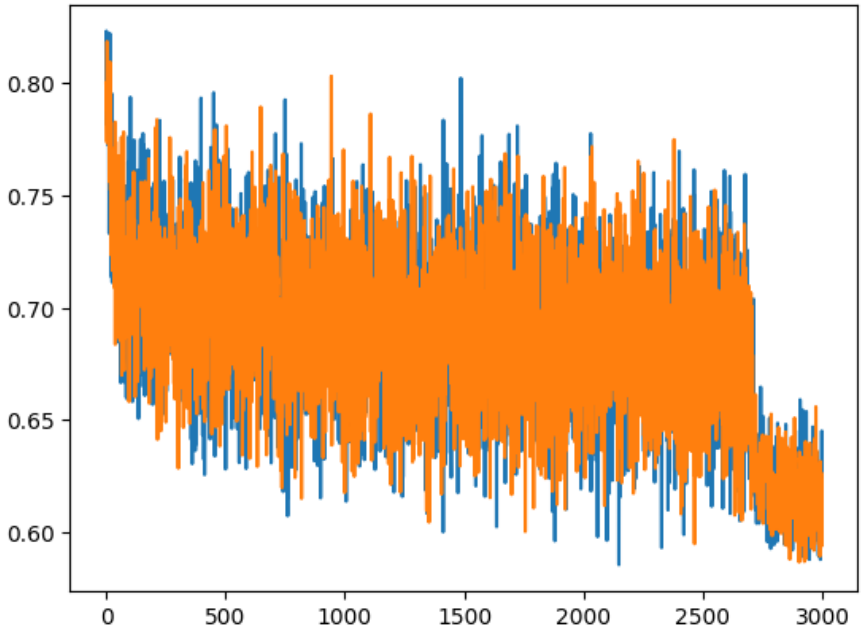
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.818497648427585, 0.5868535143066557)
```

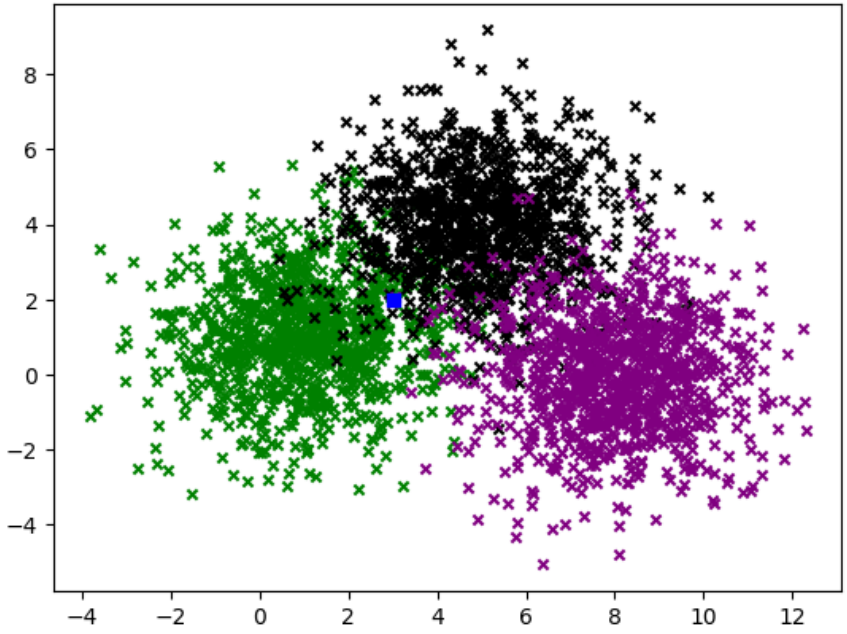
```
print(w.numpy())
print(b.numpy())
```

```
[[-0.34671909  0.45493384  0.99923588]
 [ 0.50051065  1.28724549 -0.12440768]]
[ 3.19977748 -0.61944005 -1.24581024]
```

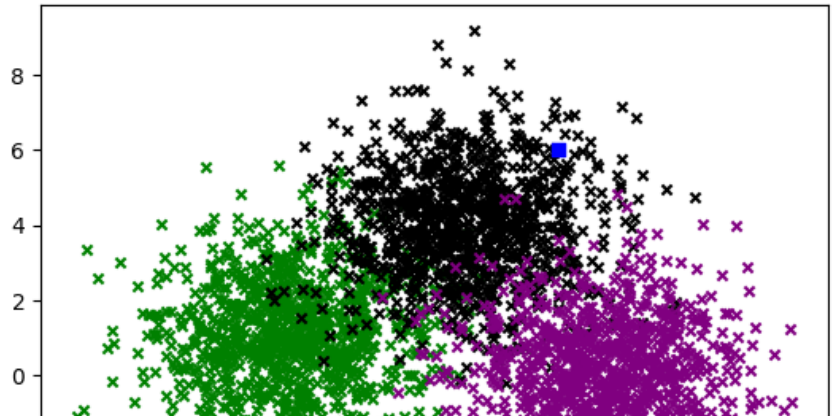
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



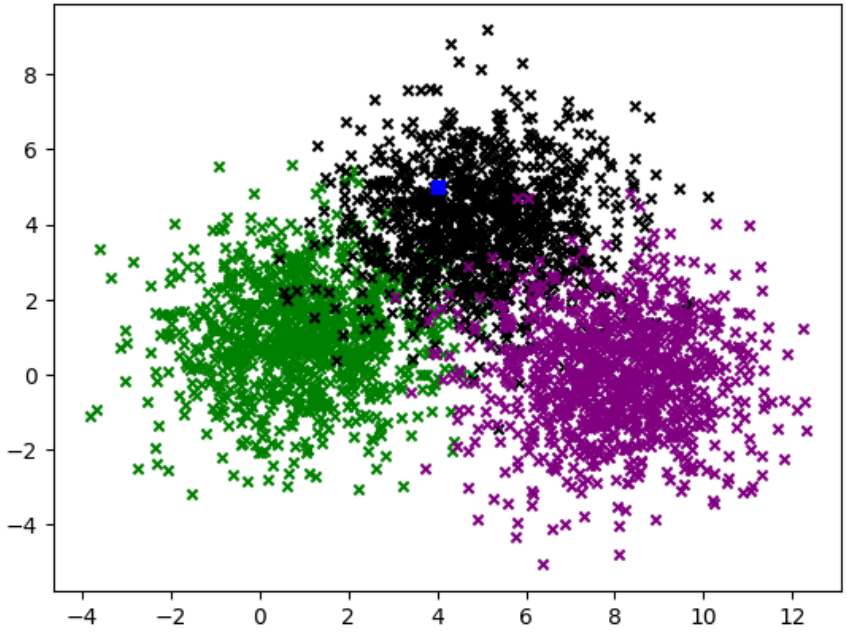
```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)


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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, 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(0.5835849893192871, shape=(), dtype=float64) val_loss tf.Tensor(0.6058292489788952, shape=(), dtype=float64)
loss tf.Tensor(0.6042925869427216, shape=(), dtype=float64) val_loss tf.Tensor(0.6015325423499391, shape=(), dtype=float64)
loss tf.Tensor(0.5972043923197844, shape=(), dtype=float64) val_loss tf.Tensor(0.613172722460744, shape=(), dtype=float64)
loss tf.Tensor(0.5777290557242692, shape=(), dtype=float64) val_loss tf.Tensor(0.5766284773857324, shape=(), dtype=float64)
loss tf.Tensor(0.6234018533813865, shape=(), dtype=float64) val_loss tf.Tensor(0.6099814298729087, shape=(), dtype=float64)
loss tf.Tensor(0.6507988240104031, shape=(), dtype=float64) val_loss tf.Tensor(0.5877696225711115, shape=(), dtype=float64)
loss tf.Tensor(0.5961222488053337, shape=(), dtype=float64) val_loss tf.Tensor(0.6117132328509672, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.8596188835808027, 0.5684671323075976)
```

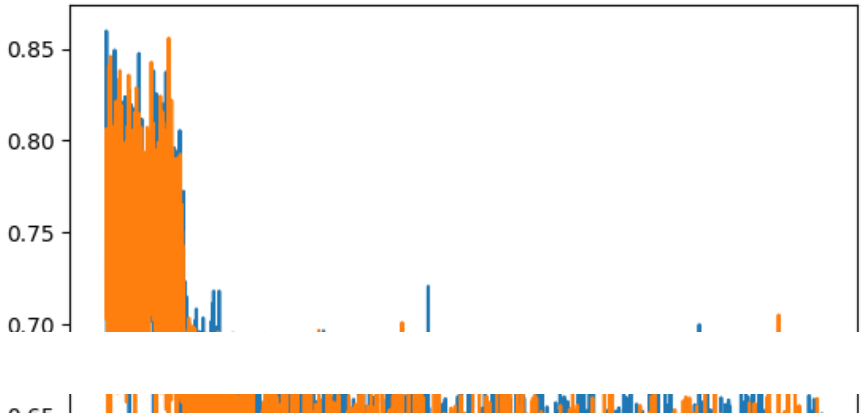
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.8557337843902377, 0.5689930781802317)
```

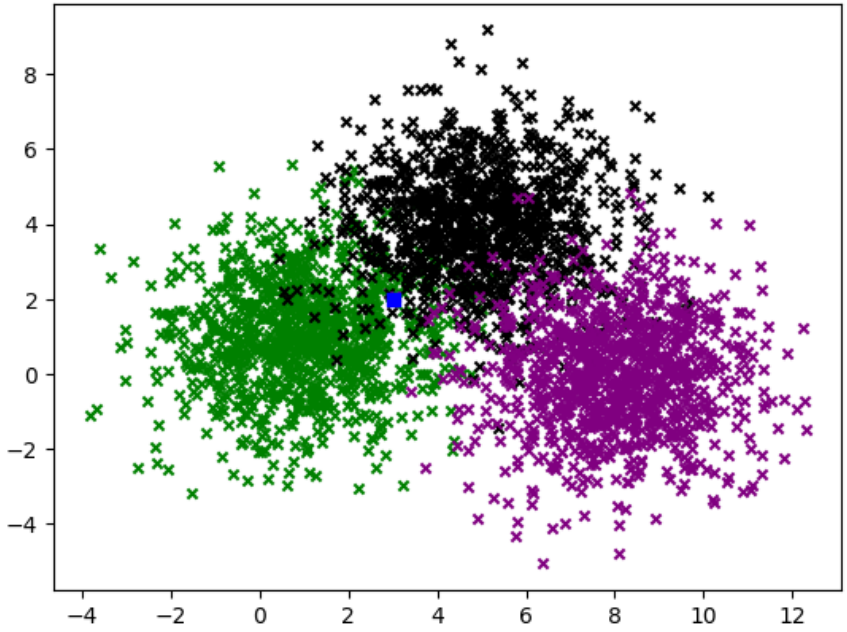
```
print(w.numpy())
print(b.numpy())
```

```
[[-0.37357033  0.6104931   1.33421813]
 [ 0.39449509  1.70300704 -0.52705518]]
[ 4.71998902 -1.30802162 -1.43951778]
```

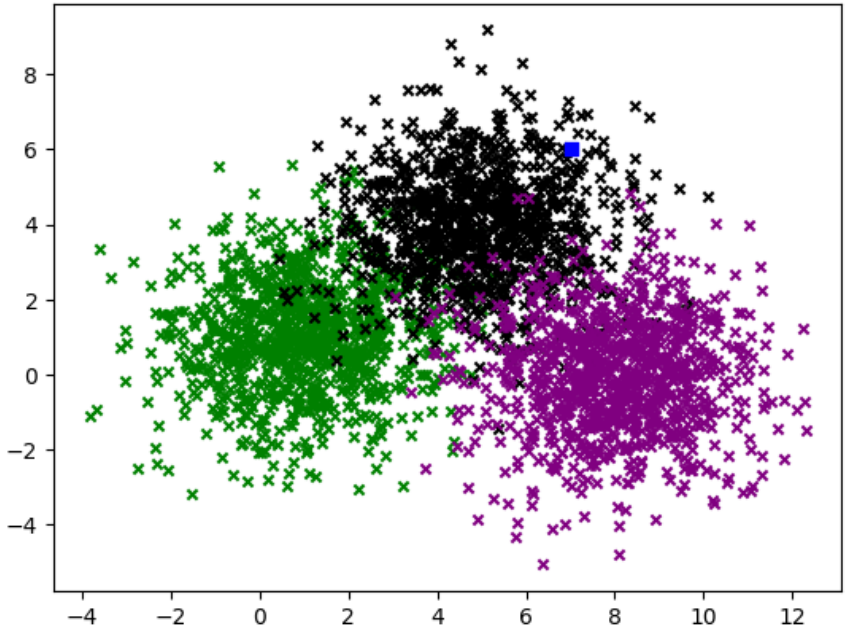
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



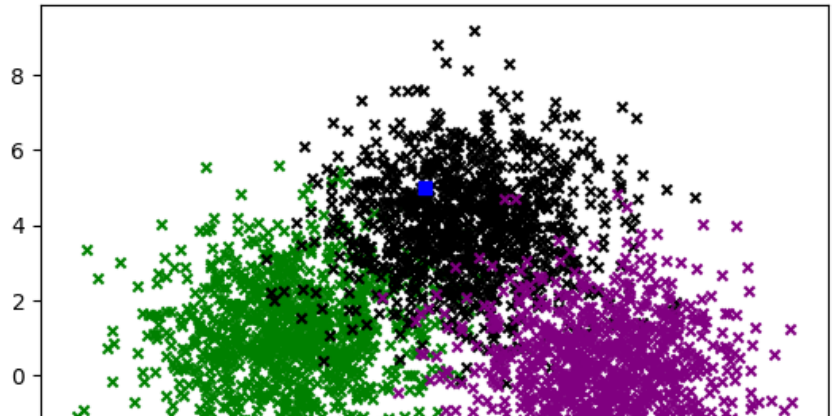

```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```



```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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, 3)))
b = tf.Variable(np.random.random((3)))
data = points
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)


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

    val_loss = tf.keras.losses.BinaryCrossentropy(from_logits=True)(labels_val_batch, pred_l_val)
    Val_loss.append(val_loss.numpy())
```

```
      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(0.5994178701473307, shape=(), dtype=float64) val_loss tf.Tensor(0.5988271627463142, shape=(), dtype=float64)
loss tf.Tensor(0.6070900220054605, shape=(), dtype=float64) val_loss tf.Tensor(0.5956633987699028, shape=(), dtype=float64)
loss tf.Tensor(0.5998079750634293, shape=(), dtype=float64) val_loss tf.Tensor(0.6108227302209801, shape=(), dtype=float64)
loss tf.Tensor(0.6042203170915631, shape=(), dtype=float64) val_loss tf.Tensor(0.6083168205734868, shape=(), dtype=float64)
loss tf.Tensor(0.6038216014274359, shape=(), dtype=float64) val_loss tf.Tensor(0.6004683016572127, shape=(), dtype=float64)
loss tf.Tensor(0.595714657431281, shape=(), dtype=float64) val_loss tf.Tensor(0.6035500574417516, shape=(), dtype=float64)
loss tf.Tensor(0.6076306598905138, shape=(), dtype=float64) val_loss tf.Tensor(0.5998449017433207, shape=(), dtype=float64)
loss tf.Tensor(0.6041969146693521, shape=(), dtype=float64) val_loss tf.Tensor(0.6198549064373382, shape=(), dtype=float64)
loss tf.Tensor(0.6000091759628566, shape=(), dtype=float64) val_loss tf.Tensor(0.615541138724338, shape=(), dtype=float64)
loss tf.Tensor(0.612867381818338, shape=(), dtype=float64) val_loss tf.Tensor(0.6075382444938225, shape=(), dtype=float64)
loss tf.Tensor(0.5972909415022238, shape=(), dtype=float64) val_loss tf.Tensor(0.5958455911978963, shape=(), dtype=float64)
loss tf.Tensor(0.6099155584713055, shape=(), dtype=float64) val_loss tf.Tensor(0.6108264100726667, shape=(), dtype=float64)
loss tf.Tensor(0.6055200186235681, shape=(), dtype=float64) val_loss tf.Tensor(0.6022817800865244, shape=(), dtype=float64)
loss tf.Tensor(0.6135875389242164, shape=(), dtype=float64) val_loss tf.Tensor(0.590147099489489, shape=(), dtype=float64)
loss tf.Tensor(0.602526546508439, shape=(), dtype=float64) val_loss tf.Tensor(0.5986915157230608, shape=(), dtype=float64)
loss tf.Tensor(0.600821870763268, shape=(), dtype=float64) val_loss tf.Tensor(0.6016287160972342, shape=(), dtype=float64)
loss tf.Tensor(0.6057415277284597, shape=(), dtype=float64) val_loss tf.Tensor(0.6090360554461974, shape=(), dtype=float64)
```

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

```
4800 7200 1200 1800
```

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

```
(2400, 2) (2400, 3) (600, 2) (600, 3)
```

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

```
(0.7388355828034727, 0.5878175165374722)
```

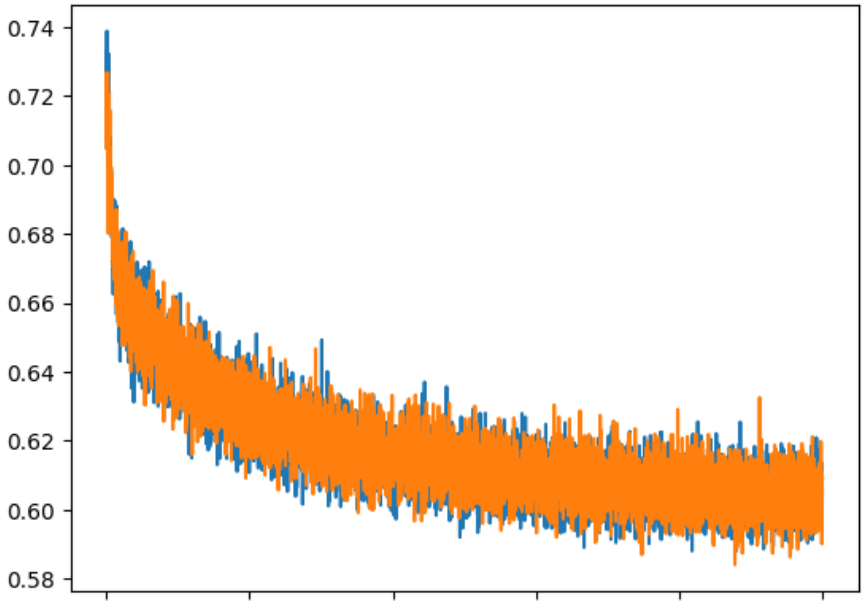
```
np.max(Val_loss),np.min(Val_loss)
```

```
(0.7267592620981006, 0.5840680254615334)
```

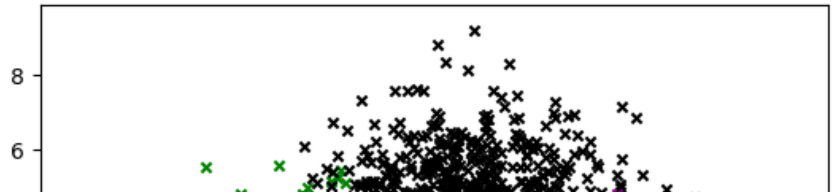
```
print(w.numpy())
print(b.numpy())
```

```
[[-0.49087696  0.50240524  1.26525779]
 [ 0.78001928  2.00482169 -0.16051308]]
[ 4.60660195 -1.33980762 -1.80413557]
```

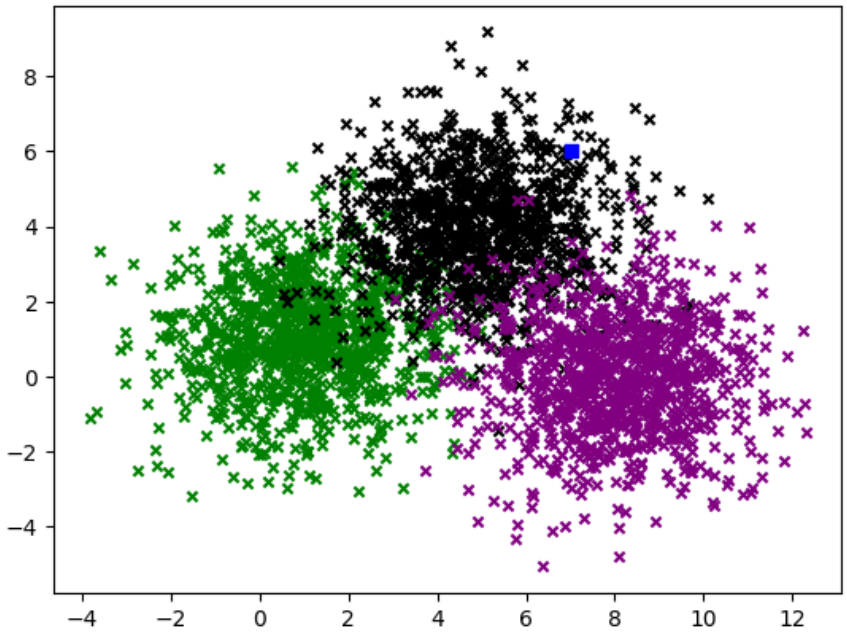
```
plt.plot(Loss)
plt.plot(Val_loss)
plt.show()
```



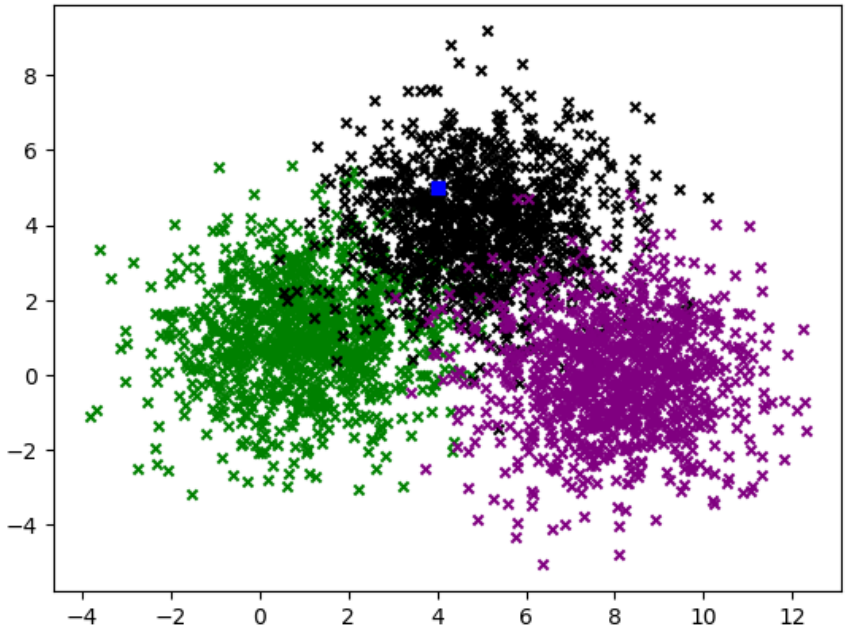

```
x=3.0
y=2.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=7.0
y=6.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', s=20)
plt.scatter(x,y,c='b', marker='s')
plt.show()
```




```
x=4.0
y=5.0
plt.scatter(x_label0, y_label0, c='green', marker='x', s=20)
plt.scatter(x_label1, y_label1, c='black', marker='x', s=20)
plt.scatter(x_label2, y_label2, c='purple', marker='x', 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 epochs of 100, a batch of 100.