Neural Network Code:

```
y_train = to_cat(t_train)
y_valid = to_cat(t_valid)
alpha = 0.001  #learning rate
nEpoch = 5000
nN = 50 # number of neurons
nF = X_train.shape[1] #number of features
nC = y_train.shape[1] #number of classes
w0 = 2*np.random.random((nF, nN)) - 1
w1 = 2*np.random.random((nN, nC)) - 1
CCR_train = np.zeros((nEpoch,1))
CCR_valid = np.zeros((nEpoch,1))
for i in range(nEpoch):
   #F
   a0 = X_train
   a1 = sigmoid_func(np.dot(a0, w0))
   a2 = sigmoid_func(np.dot(a1, w1))
   #BP
   e2 = a2 - y_train
   delt2 = e2 * deriv_sigmoid(a2)
   e1 = delt2.dot(w1.T)
   delt1 = e1 * deriv_sigmoid(a1)
   w1 -= a1.T.dot(delt2) * alpha
   w0 -= a0.T.dot(delt1) * alpha
   #FF
   tp_train = feedforward(X_train, w0, w1)
   tp_valid = feedforward(X_valid, w0, w1)
   #Evaluation
   CCR_train[i] = np.sum(tp_train.ravel() == t_train.ravel())/t_train.shape[0]
   CCR_valid[i] = np.sum(tp_valid.ravel() == t_valid.ravel())/t_valid.shape[0]
   if i%20==0:
      print('Epoch: {} and Train Acc= {}'.format(i, CCR_train[i]))
      print('Epoch: {} and Valid Acc= {}'.format(i, CCR_valid[i]))
plt.plot(CCR_train, 'b')
plt.plot(CCR_valid, 'k')
plt.xlabel('Epoch')
plt.ylabel('CCR')
plt.show()
#FF
t_train_predicted = feedforward(X_train, w0, w1)
t_valid_predicted = feedforward(X_valid, w0, w1)
t_test_predicted = feedforward(X_test, w0, w1)
#Evaluation
accuracy_train = np.sum(t_train_predicted.ravel() == t_train.ravel())/t_train.shape[0]
accuracy_valid = np.sum(t_valid_predicted.ravel() == t_valid.ravel())/t_valid.shape[0]
f1_score_train = f1_score(t_train_predicted.ravel(), t_train.ravel(), average='weighted')
f1_score_valid = f1_score(t_valid_predicted.ravel(), t_valid.ravel(), average='weighted')
```