Introduction

Welcome to this lab! At this lab, we will learn how to implement Graph Convolutional Networks (GCN) based on the paper GCN.

Exercise

Download data and install packages

In [ ]:
```
!gdown --id "1Z748ksOFWZ8WLsy0P8T7eYA6XLVLieX3&export=download"
!unrar x -Y "/content/lab2.rar" -d "/content/"
```

```
Downloading...
From: https://drive.google.com/uc?id=1Z748ksOFWZ8WLsy0P8T7eYA6XLVLieX3&export=
download
To: /content/lab2.rar
100% 599/599 [00:00<00:00, 1.65MB/s]

UNRAR 5.50 freeware      Copyright (c) 1993-2017 Alexander Roshal


Extracting from /content/lab2.rar

Extracting  /content/lab2_edgelist.txt                                55%
OK
Extracting  /content/lab2_attributes.csv                              89%
OK
All OK
```

Packages: Import neccesary packages

In [ ]:
```
import torch
import torch.nn as nn
import numpy as np
import networkx as nx
import torch.optim as optim
import torch.nn.functional as F
import pandas as pd
```

Utils: Processing data

In [ ]:
```
def create_graphs_with_attributes(edgelist_filepath,attributes_filepath):
    graph=nx.read_edgelist(edgelist_filepath,nodetype=int)
    attributes=pd.read_csv(attributes_filepath,index_col=['node'])
    att_values = {a:{'role':b[0],'community':b[1]} for a,b in enumerate(attril
    nx.set_node_attributes(graph,att_values)
    return graph


def create_train_test(graph):
    X_train,Y_train,X_test,Y_test=[],[],[],[]
    for node, data in graph.nodes(data=True):
        if data['role'] in ['Administrator','Instructor']:
            X_train.append(node)
            Y_train.append(data['role']=='Administrator')
```

```
        elif data['role'] =='Member':
            X_test.append(node)
            Y_test.append(data['community']=='Administrator')
    return np.asarray(X_train),np.asarray(Y_train),np.asarray(X_test),np.asar

def create_features(graph): # create input features, concatenation of identit
    A = nx.to_numpy_matrix(graph)
    X_1 = torch.eye(A.shape[0]) # identity matrix
    X_2 = torch.zeros((A.shape[0], 2)) # shortest path to the targets as 2nd
    node_distance_instructor = nx.shortest_path_length(graph, target=33)
    node_distance_administrator = nx.shortest_path_length(graph, target=0)

    for node in graph.nodes():
        X_2[node][0] = node_distance_administrator[node]
        X_2[node][1] = node_distance_instructor[node]

    return torch.cat((X_1,X_2),dim=1)
```

TO DO: Implement Graph Convolutional Networks (GCN)

Method

- $\hat{A} = \tilde{D}^{\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$
- Filter layer 1: $(X, A) \rightarrow \hat{A}\, X\, W^{(0)}$
- Filter layer 2: $\hat{A}\, X\, W^{(0)} \rightarrow \hat{A} \times ReLU\left(\hat{A}XW^{(0)}\right)\times W^{(1)}$
- Output layer: $\hat{A} \times ReLU\left(\hat{A}XW^{(0)}\right)\times W^{(1)} \rightarrow Softmax\left(\hat{A} \times ReLU\left(\hat{A}XW^{(0)}\right)\times W^{(1)}\right)$

In [ ]:

```
from torch.nn.parameter import Parameter

class GCN_layer(nn.Module):
    """
      Define filter layer 1/2 like in the above image
      Calculate A_hat first then,
      Input: adj_matrix with input features X
    """

    def __init__(self, inputs_shape, outputs_shape):
        super(GCN_layer, self).__init__()

        self.W=Parameter(torch.rand(inputs_shape,outputs_shape),requires_grad
        self.bias = Parameter(torch.rand(outputs_shape),requires_grad=True)


    def forward(self, adj_matrix, input_features):
        # TO DO: Calculate A_hat
        aggregate=torch.mm(A_hat,input_features)
        propagate=torch.mm(aggregate,self.W)+self.bias
        return propagate
```

In [ ]:

```
class GCN(nn.Module):


    def __init__(self,inputs_shape,outputs_shape,n_classes,activation='Relu')
        super(GCN, self).__init__()
```

```
        self.layer1=GCN_layer(inputs_shape,outputs_shape)
        self.layer2=GCN_layer(outputs_shape,n_classes)


        if activation =='Tanh':
            self.activation = nn.Tanh()
        elif activation=='Sigmoid':
            self.activation = nn.Sigmoid()
        elif activation=='Softmax':
            self.activation=nn.Softmax()
        elif activation=='Relu':
            self.activation=nn.ReLU()

        self.softmax=nn.Softmax()


    def forward(self,adj_matrix,input_features):
        # TO DO: Define output for each layer. Please look at the method in t

        return x
```

In [ ]:

```
import torch
from sklearn.metrics import classification_report
import torch.optim as optim
import torch.nn.functional as F
import numpy as np
import networkx as nx


graph=create_graphs_with_attributes('lab2_edgelist.txt','lab2_attributes.csv'
A = np.array(nx.to_numpy_matrix(graph)) # adjadjency matrix

class Trainer():
    def __init__(self,model,optimizer,loss_function,epochs):

        self.model=model
        self.optimizer=optimizer
        self.loss_function=loss_function
        self.epochs=epochs

    def train(self,X_train,Y_train):

        y_train=torch.from_numpy(Y_train.astype(int)).type(torch.LongTensor)
        tot_loss=0.0
        all_preds=[]

        for t in range(self.epochs):
            epoch_loss = 0.0

            #model.train()
            y_pred=self.model(A, create_features(graph))
            all_preds.append(y_pred)
            loss = self.loss_function(y_pred[X_train], y_train)
            self.optimizer.zero_grad()
            epoch_loss+=loss
            tot_loss+=loss
            loss.backward()
            self.optimizer.step()

            print(str(t),'epoch_loss:'+str(epoch_loss),'total loss:'+str(tot_
```

```
        self.all_preds=all_preds

    def test(self,X_test,Y_test):

        self.model.eval()
        y_test=torch.from_numpy(Y_test.astype(int)).type(torch.LongTensor)
        y_pred=self.all_preds[-1]  # preds of last epoch
        loss_test = self.loss_function(y_pred[X_test],y_test)
        print('validation loss is equal to: '+str(loss_test))

    def visualize_classification(self,graph,Y_test,classification):
        last_epoch = self.all_preds[self.epochs-1].detach().numpy() # get out
        predicted_class = np.argmax(last_epoch, axis=-1) # take the unit with
        color = np.where(predicted_class==0, 'c', 'r')
        pos = nx.kamada_kawai_layout(graph)
        nx.draw_networkx(graph, pos, node_color=color, with_labels=True, node
        if classification==True:
            print(classification_report(predicted_class[1:-1],Y_test))
```

Run GCN

In [ ]:

```
graph=create_graphs_with_attributes('lab2_edgelist.txt','lab2_attributes.csv'
A = np.array(nx.to_numpy_matrix(graph)) # adjadjency matrix
X_train,Y_train,X_test,Y_test= create_train_test(graph)

model=GCN(inputs_shape=create_features(graph).shape[1],outputs_shape=4,n_clas
trainer = Trainer(
    model,
    optimizer = optim.Adam(model.parameters(), lr=0.01),
    loss_function=F.cross_entropy,
    epochs=250
)

trainer.train(X_train,Y_train)
trainer.test(X_test,Y_test)
trainer.visualize_classification(graph,Y_test,classification=True)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:28: UserWarning:
Implicit dimension choice for softmax has been deprecated. Change the call to
include dim=X as an argument.
0 epoch_loss:tensor(0.7081, grad_fn=<AddBackward0>) total loss:tensor(0.7081,
grad_fn=<AddBackward0>)
1 epoch_loss:tensor(0.7062, grad_fn=<AddBackward0>) total loss:tensor(1.4143,
grad_fn=<AddBackward0>)
2 epoch_loss:tensor(0.7043, grad_fn=<AddBackward0>) total loss:tensor(2.1186,
grad_fn=<AddBackward0>)
3 epoch_loss:tensor(0.7025, grad_fn=<AddBackward0>) total loss:tensor(2.8211,
grad_fn=<AddBackward0>)
4 epoch_loss:tensor(0.7007, grad_fn=<AddBackward0>) total loss:tensor(3.5218,
grad_fn=<AddBackward0>)
5 epoch_loss:tensor(0.6990, grad_fn=<AddBackward0>) total loss:tensor(4.2208,
grad_fn=<AddBackward0>)
6 epoch_loss:tensor(0.6973, grad_fn=<AddBackward0>) total loss:tensor(4.9181,
grad_fn=<AddBackward0>)
7 epoch_loss:tensor(0.6957, grad_fn=<AddBackward0>) total loss:tensor(5.6139,
grad_fn=<AddBackward0>)
8 epoch_loss:tensor(0.6942, grad_fn=<AddBackward0>) total loss:tensor(6.3080,
grad_fn=<AddBackward0>)
9 epoch_loss:tensor(0.6926, grad_fn=<AddBackward0>) total loss:tensor(7.0007,
grad_fn=<AddBackward0>)
10 epoch_loss:tensor(0.6912, grad_fn=<AddBackward0>) total loss:tensor(7.6918,
grad_fn=<AddBackward0>)
11 epoch_loss:tensor(0.6898, grad_fn=<AddBackward0>) total loss:tensor(8.3816,
```

```
grad_fn=<AddBackward0>)
12 epoch_loss:tensor(0.6884, grad_fn=<AddBackward0>) total loss:tensor(9.0700,
grad_fn=<AddBackward0>)
13 epoch_loss:tensor(0.6871, grad_fn=<AddBackward0>) total loss:tensor(9.7571,
grad_fn=<AddBackward0>)
14 epoch_loss:tensor(0.6858, grad_fn=<AddBackward0>) total loss:tensor(10.442
9, grad_fn=<AddBackward0>)
15 epoch_loss:tensor(0.6846, grad_fn=<AddBackward0>) total loss:tensor(11.127
5, grad_fn=<AddBackward0>)
16 epoch_loss:tensor(0.6835, grad_fn=<AddBackward0>) total loss:tensor(11.811
0, grad_fn=<AddBackward0>)
17 epoch_loss:tensor(0.6824, grad_fn=<AddBackward0>) total loss:tensor(12.493
4, grad_fn=<AddBackward0>)
18 epoch_loss:tensor(0.6813, grad_fn=<AddBackward0>) total loss:tensor(13.174
8, grad_fn=<AddBackward0>)
19 epoch_loss:tensor(0.6803, grad_fn=<AddBackward0>) total loss:tensor(13.855
1, grad_fn=<AddBackward0>)
20 epoch_loss:tensor(0.6794, grad_fn=<AddBackward0>) total loss:tensor(14.534
5, grad_fn=<AddBackward0>)
21 epoch_loss:tensor(0.6784, grad_fn=<AddBackward0>) total loss:tensor(15.212
9, grad_fn=<AddBackward0>)
22 epoch_loss:tensor(0.6775, grad_fn=<AddBackward0>) total loss:tensor(15.890
4, grad_fn=<AddBackward0>)
23 epoch_loss:tensor(0.6767, grad_fn=<AddBackward0>) total loss:tensor(16.567
1, grad_fn=<AddBackward0>)
24 epoch_loss:tensor(0.6758, grad_fn=<AddBackward0>) total loss:tensor(17.242
9, grad_fn=<AddBackward0>)
25 epoch_loss:tensor(0.6749, grad_fn=<AddBackward0>) total loss:tensor(17.917
8, grad_fn=<AddBackward0>)
26 epoch_loss:tensor(0.6741, grad_fn=<AddBackward0>) total loss:tensor(18.591
9, grad_fn=<AddBackward0>)
27 epoch_loss:tensor(0.6732, grad_fn=<AddBackward0>) total loss:tensor(19.265
0, grad_fn=<AddBackward0>)
28 epoch_loss:tensor(0.6723, grad_fn=<AddBackward0>) total loss:tensor(19.937
3, grad_fn=<AddBackward0>)
29 epoch_loss:tensor(0.6714, grad_fn=<AddBackward0>) total loss:tensor(20.608
7, grad_fn=<AddBackward0>)
30 epoch_loss:tensor(0.6704, grad_fn=<AddBackward0>) total loss:tensor(21.279
1, grad_fn=<AddBackward0>)
31 epoch_loss:tensor(0.6694, grad_fn=<AddBackward0>) total loss:tensor(21.948
5, grad_fn=<AddBackward0>)
32 epoch_loss:tensor(0.6684, grad_fn=<AddBackward0>) total loss:tensor(22.616
9, grad_fn=<AddBackward0>)
33 epoch_loss:tensor(0.6673, grad_fn=<AddBackward0>) total loss:tensor(23.284
2, grad_fn=<AddBackward0>)
34 epoch_loss:tensor(0.6662, grad_fn=<AddBackward0>) total loss:tensor(23.950
3, grad_fn=<AddBackward0>)
35 epoch_loss:tensor(0.6650, grad_fn=<AddBackward0>) total loss:tensor(24.615
3, grad_fn=<AddBackward0>)
36 epoch_loss:tensor(0.6638, grad_fn=<AddBackward0>) total loss:tensor(25.279
1, grad_fn=<AddBackward0>)
37 epoch_loss:tensor(0.6625, grad_fn=<AddBackward0>) total loss:tensor(25.941
5, grad_fn=<AddBackward0>)
38 epoch_loss:tensor(0.6611, grad_fn=<AddBackward0>) total loss:tensor(26.602
7, grad_fn=<AddBackward0>)
39 epoch_loss:tensor(0.6598, grad_fn=<AddBackward0>) total loss:tensor(27.262
4, grad_fn=<AddBackward0>)
40 epoch_loss:tensor(0.6583, grad_fn=<AddBackward0>) total loss:tensor(27.920
7, grad_fn=<AddBackward0>)
41 epoch_loss:tensor(0.6568, grad_fn=<AddBackward0>) total loss:tensor(28.577
6, grad_fn=<AddBackward0>)
42 epoch_loss:tensor(0.6553, grad_fn=<AddBackward0>) total loss:tensor(29.232
8, grad_fn=<AddBackward0>)
43 epoch_loss:tensor(0.6537, grad_fn=<AddBackward0>) total loss:tensor(29.886
5, grad_fn=<AddBackward0>)
44 epoch_loss:tensor(0.6521, grad_fn=<AddBackward0>) total loss:tensor(30.538
6, grad_fn=<AddBackward0>)
45 epoch_loss:tensor(0.6504, grad_fn=<AddBackward0>) total loss:tensor(31.189
0, grad_fn=<AddBackward0>)
```

```
46 epoch_loss:tensor(0.6486, grad_fn=<AddBackward0>) total loss:tensor(31.837
6, grad_fn=<AddBackward0>)
47 epoch_loss:tensor(0.6469, grad_fn=<AddBackward0>) total loss:tensor(32.484
5, grad_fn=<AddBackward0>)
48 epoch_loss:tensor(0.6451, grad_fn=<AddBackward0>) total loss:tensor(33.129
6, grad_fn=<AddBackward0>)
49 epoch_loss:tensor(0.6432, grad_fn=<AddBackward0>) total loss:tensor(33.772
7, grad_fn=<AddBackward0>)
50 epoch_loss:tensor(0.6413, grad_fn=<AddBackward0>) total loss:tensor(34.414
0, grad_fn=<AddBackward0>)
51 epoch_loss:tensor(0.6393, grad_fn=<AddBackward0>) total loss:tensor(35.053
3, grad_fn=<AddBackward0>)
52 epoch_loss:tensor(0.6373, grad_fn=<AddBackward0>) total loss:tensor(35.690
7, grad_fn=<AddBackward0>)
53 epoch_loss:tensor(0.6353, grad_fn=<AddBackward0>) total loss:tensor(36.325
9, grad_fn=<AddBackward0>)
54 epoch_loss:tensor(0.6332, grad_fn=<AddBackward0>) total loss:tensor(36.959
1, grad_fn=<AddBackward0>)
55 epoch_loss:tensor(0.6310, grad_fn=<AddBackward0>) total loss:tensor(37.590
1, grad_fn=<AddBackward0>)
56 epoch_loss:tensor(0.6288, grad_fn=<AddBackward0>) total loss:tensor(38.218
9, grad_fn=<AddBackward0>)
57 epoch_loss:tensor(0.6266, grad_fn=<AddBackward0>) total loss:tensor(38.845
5, grad_fn=<AddBackward0>)
58 epoch_loss:tensor(0.6243, grad_fn=<AddBackward0>) total loss:tensor(39.469
8, grad_fn=<AddBackward0>)
59 epoch_loss:tensor(0.6219, grad_fn=<AddBackward0>) total loss:tensor(40.091
7, grad_fn=<AddBackward0>)
60 epoch_loss:tensor(0.6195, grad_fn=<AddBackward0>) total loss:tensor(40.711
2, grad_fn=<AddBackward0>)
61 epoch_loss:tensor(0.6171, grad_fn=<AddBackward0>) total loss:tensor(41.328
3, grad_fn=<AddBackward0>)
62 epoch_loss:tensor(0.6146, grad_fn=<AddBackward0>) total loss:tensor(41.942
9, grad_fn=<AddBackward0>)
63 epoch_loss:tensor(0.6120, grad_fn=<AddBackward0>) total loss:tensor(42.554
9, grad_fn=<AddBackward0>)
64 epoch_loss:tensor(0.6094, grad_fn=<AddBackward0>) total loss:tensor(43.164
3, grad_fn=<AddBackward0>)
65 epoch_loss:tensor(0.6068, grad_fn=<AddBackward0>) total loss:tensor(43.771
1, grad_fn=<AddBackward0>)
66 epoch_loss:tensor(0.6041, grad_fn=<AddBackward0>) total loss:tensor(44.375
3, grad_fn=<AddBackward0>)
67 epoch_loss:tensor(0.6014, grad_fn=<AddBackward0>) total loss:tensor(44.976
7, grad_fn=<AddBackward0>)
68 epoch_loss:tensor(0.5986, grad_fn=<AddBackward0>) total loss:tensor(45.575
3, grad_fn=<AddBackward0>)
69 epoch_loss:tensor(0.5958, grad_fn=<AddBackward0>) total loss:tensor(46.171
1, grad_fn=<AddBackward0>)
70 epoch_loss:tensor(0.5930, grad_fn=<AddBackward0>) total loss:tensor(46.764
1, grad_fn=<AddBackward0>)
71 epoch_loss:tensor(0.5901, grad_fn=<AddBackward0>) total loss:tensor(47.354
2, grad_fn=<AddBackward0>)
72 epoch_loss:tensor(0.5872, grad_fn=<AddBackward0>) total loss:tensor(47.941
4, grad_fn=<AddBackward0>)
73 epoch_loss:tensor(0.5842, grad_fn=<AddBackward0>) total loss:tensor(48.525
6, grad_fn=<AddBackward0>)
74 epoch_loss:tensor(0.5812, grad_fn=<AddBackward0>) total loss:tensor(49.106
9, grad_fn=<AddBackward0>)
75 epoch_loss:tensor(0.5782, grad_fn=<AddBackward0>) total loss:tensor(49.685
1, grad_fn=<AddBackward0>)
76 epoch_loss:tensor(0.5752, grad_fn=<AddBackward0>) total loss:tensor(50.260
3, grad_fn=<AddBackward0>)
77 epoch_loss:tensor(0.5721, grad_fn=<AddBackward0>) total loss:tensor(50.832
3, grad_fn=<AddBackward0>)
78 epoch_loss:tensor(0.5689, grad_fn=<AddBackward0>) total loss:tensor(51.401
3, grad_fn=<AddBackward0>)
79 epoch_loss:tensor(0.5658, grad_fn=<AddBackward0>) total loss:tensor(51.967
1, grad_fn=<AddBackward0>)
80 epoch_loss:tensor(0.5626, grad_fn=<AddBackward0>) total loss:tensor(52.529
```

```
7, grad_fn=<AddBackward0>)
81 epoch_loss:tensor(0.5594, grad_fn=<AddBackward0>) total loss:tensor(53.089
2, grad_fn=<AddBackward0>)
82 epoch_loss:tensor(0.5562, grad_fn=<AddBackward0>) total loss:tensor(53.645
4, grad_fn=<AddBackward0>)
83 epoch_loss:tensor(0.5530, grad_fn=<AddBackward0>) total loss:tensor(54.198
4, grad_fn=<AddBackward0>)
84 epoch_loss:tensor(0.5498, grad_fn=<AddBackward0>) total loss:tensor(54.748
2, grad_fn=<AddBackward0>)
85 epoch_loss:tensor(0.5465, grad_fn=<AddBackward0>) total loss:tensor(55.294
7, grad_fn=<AddBackward0>)
86 epoch_loss:tensor(0.5432, grad_fn=<AddBackward0>) total loss:tensor(55.837
9, grad_fn=<AddBackward0>)
87 epoch_loss:tensor(0.5400, grad_fn=<AddBackward0>) total loss:tensor(56.377
9, grad_fn=<AddBackward0>)
88 epoch_loss:tensor(0.5367, grad_fn=<AddBackward0>) total loss:tensor(56.914
6, grad_fn=<AddBackward0>)
89 epoch_loss:tensor(0.5334, grad_fn=<AddBackward0>) total loss:tensor(57.448
0, grad_fn=<AddBackward0>)
90 epoch_loss:tensor(0.5301, grad_fn=<AddBackward0>) total loss:tensor(57.978
0, grad_fn=<AddBackward0>)
91 epoch_loss:tensor(0.5268, grad_fn=<AddBackward0>) total loss:tensor(58.504
8, grad_fn=<AddBackward0>)
92 epoch_loss:tensor(0.5235, grad_fn=<AddBackward0>) total loss:tensor(59.028
3, grad_fn=<AddBackward0>)
93 epoch_loss:tensor(0.5202, grad_fn=<AddBackward0>) total loss:tensor(59.548
5, grad_fn=<AddBackward0>)
94 epoch_loss:tensor(0.5169, grad_fn=<AddBackward0>) total loss:tensor(60.065
4, grad_fn=<AddBackward0>)
95 epoch_loss:tensor(0.5136, grad_fn=<AddBackward0>) total loss:tensor(60.579
0, grad_fn=<AddBackward0>)
96 epoch_loss:tensor(0.5104, grad_fn=<AddBackward0>) total loss:tensor(61.089
4, grad_fn=<AddBackward0>)
97 epoch_loss:tensor(0.5071, grad_fn=<AddBackward0>) total loss:tensor(61.596
5, grad_fn=<AddBackward0>)
98 epoch_loss:tensor(0.5039, grad_fn=<AddBackward0>) total loss:tensor(62.100
4, grad_fn=<AddBackward0>)
99 epoch_loss:tensor(0.5006, grad_fn=<AddBackward0>) total loss:tensor(62.601
0, grad_fn=<AddBackward0>)
100 epoch_loss:tensor(0.4974, grad_fn=<AddBackward0>) total loss:tensor(63.098
4, grad_fn=<AddBackward0>)
101 epoch_loss:tensor(0.4942, grad_fn=<AddBackward0>) total loss:tensor(63.592
6, grad_fn=<AddBackward0>)
102 epoch_loss:tensor(0.4911, grad_fn=<AddBackward0>) total loss:tensor(64.083
7, grad_fn=<AddBackward0>)
103 epoch_loss:tensor(0.4880, grad_fn=<AddBackward0>) total loss:tensor(64.571
7, grad_fn=<AddBackward0>)
104 epoch_loss:tensor(0.4849, grad_fn=<AddBackward0>) total loss:tensor(65.056
6, grad_fn=<AddBackward0>)
105 epoch_loss:tensor(0.4818, grad_fn=<AddBackward0>) total loss:tensor(65.538
4, grad_fn=<AddBackward0>)
106 epoch_loss:tensor(0.4788, grad_fn=<AddBackward0>) total loss:tensor(66.017
2, grad_fn=<AddBackward0>)
107 epoch_loss:tensor(0.4758, grad_fn=<AddBackward0>) total loss:tensor(66.493
0, grad_fn=<AddBackward0>)
108 epoch_loss:tensor(0.4728, grad_fn=<AddBackward0>) total loss:tensor(66.965
8, grad_fn=<AddBackward0>)
109 epoch_loss:tensor(0.4698, grad_fn=<AddBackward0>) total loss:tensor(67.435
6, grad_fn=<AddBackward0>)
110 epoch_loss:tensor(0.4669, grad_fn=<AddBackward0>) total loss:tensor(67.902
5, grad_fn=<AddBackward0>)
111 epoch_loss:tensor(0.4640, grad_fn=<AddBackward0>) total loss:tensor(68.366
6, grad_fn=<AddBackward0>)
112 epoch_loss:tensor(0.4612, grad_fn=<AddBackward0>) total loss:tensor(68.827
7, grad_fn=<AddBackward0>)
113 epoch_loss:tensor(0.4583, grad_fn=<AddBackward0>) total loss:tensor(69.286
1, grad_fn=<AddBackward0>)
114 epoch_loss:tensor(0.4556, grad_fn=<AddBackward0>) total loss:tensor(69.741
6, grad_fn=<AddBackward0>)
```

```
115 epoch_loss:tensor(0.4528, grad_fn=<AddBackward0>) total loss:tensor(70.194
4, grad_fn=<AddBackward0>)
116 epoch_loss:tensor(0.4501, grad_fn=<AddBackward0>) total loss:tensor(70.644
6, grad_fn=<AddBackward0>)
117 epoch_loss:tensor(0.4475, grad_fn=<AddBackward0>) total loss:tensor(71.092
1, grad_fn=<AddBackward0>)
118 epoch_loss:tensor(0.4449, grad_fn=<AddBackward0>) total loss:tensor(71.536
9, grad_fn=<AddBackward0>)
119 epoch_loss:tensor(0.4423, grad_fn=<AddBackward0>) total loss:tensor(71.979
2, grad_fn=<AddBackward0>)
120 epoch_loss:tensor(0.4398, grad_fn=<AddBackward0>) total loss:tensor(72.419
0, grad_fn=<AddBackward0>)
121 epoch_loss:tensor(0.4373, grad_fn=<AddBackward0>) total loss:tensor(72.856
3, grad_fn=<AddBackward0>)
122 epoch_loss:tensor(0.4348, grad_fn=<AddBackward0>) total loss:tensor(73.291
1, grad_fn=<AddBackward0>)
123 epoch_loss:tensor(0.4324, grad_fn=<AddBackward0>) total loss:tensor(73.723
5, grad_fn=<AddBackward0>)
124 epoch_loss:tensor(0.4301, grad_fn=<AddBackward0>) total loss:tensor(74.153
6, grad_fn=<AddBackward0>)
125 epoch_loss:tensor(0.4278, grad_fn=<AddBackward0>) total loss:tensor(74.581
4, grad_fn=<AddBackward0>)
126 epoch_loss:tensor(0.4255, grad_fn=<AddBackward0>) total loss:tensor(75.006
9, grad_fn=<AddBackward0>)
127 epoch_loss:tensor(0.4233, grad_fn=<AddBackward0>) total loss:tensor(75.430
2, grad_fn=<AddBackward0>)
128 epoch_loss:tensor(0.4211, grad_fn=<AddBackward0>) total loss:tensor(75.851
2, grad_fn=<AddBackward0>)
129 epoch_loss:tensor(0.4189, grad_fn=<AddBackward0>) total loss:tensor(76.270
2, grad_fn=<AddBackward0>)
130 epoch_loss:tensor(0.4168, grad_fn=<AddBackward0>) total loss:tensor(76.687
0, grad_fn=<AddBackward0>)
131 epoch_loss:tensor(0.4148, grad_fn=<AddBackward0>) total loss:tensor(77.101
8, grad_fn=<AddBackward0>)
132 epoch_loss:tensor(0.4128, grad_fn=<AddBackward0>) total loss:tensor(77.514
5, grad_fn=<AddBackward0>)
133 epoch_loss:tensor(0.4108, grad_fn=<AddBackward0>) total loss:tensor(77.925
3, grad_fn=<AddBackward0>)
134 epoch_loss:tensor(0.4089, grad_fn=<AddBackward0>) total loss:tensor(78.334
2, grad_fn=<AddBackward0>)
135 epoch_loss:tensor(0.4070, grad_fn=<AddBackward0>) total loss:tensor(78.741
2, grad_fn=<AddBackward0>)
136 epoch_loss:tensor(0.4051, grad_fn=<AddBackward0>) total loss:tensor(79.146
3, grad_fn=<AddBackward0>)
137 epoch_loss:tensor(0.4033, grad_fn=<AddBackward0>) total loss:tensor(79.549
6, grad_fn=<AddBackward0>)
138 epoch_loss:tensor(0.4015, grad_fn=<AddBackward0>) total loss:tensor(79.951
1, grad_fn=<AddBackward0>)
139 epoch_loss:tensor(0.3998, grad_fn=<AddBackward0>) total loss:tensor(80.350
9, grad_fn=<AddBackward0>)
140 epoch_loss:tensor(0.3981, grad_fn=<AddBackward0>) total loss:tensor(80.749
0, grad_fn=<AddBackward0>)
141 epoch_loss:tensor(0.3964, grad_fn=<AddBackward0>) total loss:tensor(81.145
4, grad_fn=<AddBackward0>)
142 epoch_loss:tensor(0.3948, grad_fn=<AddBackward0>) total loss:tensor(81.540
2, grad_fn=<AddBackward0>)
143 epoch_loss:tensor(0.3932, grad_fn=<AddBackward0>) total loss:tensor(81.933
5, grad_fn=<AddBackward0>)
144 epoch_loss:tensor(0.3917, grad_fn=<AddBackward0>) total loss:tensor(82.325
1, grad_fn=<AddBackward0>)
145 epoch_loss:tensor(0.3902, grad_fn=<AddBackward0>) total loss:tensor(82.715
3, grad_fn=<AddBackward0>)
146 epoch_loss:tensor(0.3887, grad_fn=<AddBackward0>) total loss:tensor(83.104
0, grad_fn=<AddBackward0>)
147 epoch_loss:tensor(0.3872, grad_fn=<AddBackward0>) total loss:tensor(83.491
3, grad_fn=<AddBackward0>)
148 epoch_loss:tensor(0.3859, grad_fn=<AddBackward0>) total loss:tensor(83.877
1, grad_fn=<AddBackward0>)
149 epoch_loss:tensor(0.3845, grad_fn=<AddBackward0>) total loss:tensor(84.261
```

```
6, grad_fn=<AddBackward0>)
150 epoch_loss:tensor(0.3831, grad_fn=<AddBackward0>) total loss:tensor(84.644
7, grad_fn=<AddBackward0>)
151 epoch_loss:tensor(0.3818, grad_fn=<AddBackward0>) total loss:tensor(85.026
5, grad_fn=<AddBackward0>)
152 epoch_loss:tensor(0.3805, grad_fn=<AddBackward0>) total loss:tensor(85.407
0, grad_fn=<AddBackward0>)
153 epoch_loss:tensor(0.3793, grad_fn=<AddBackward0>) total loss:tensor(85.786
3, grad_fn=<AddBackward0>)
154 epoch_loss:tensor(0.3781, grad_fn=<AddBackward0>) total loss:tensor(86.164
4, grad_fn=<AddBackward0>)
155 epoch_loss:tensor(0.3769, grad_fn=<AddBackward0>) total loss:tensor(86.541
2, grad_fn=<AddBackward0>)
156 epoch_loss:tensor(0.3757, grad_fn=<AddBackward0>) total loss:tensor(86.916
9, grad_fn=<AddBackward0>)
157 epoch_loss:tensor(0.3745, grad_fn=<AddBackward0>) total loss:tensor(87.291
5, grad_fn=<AddBackward0>)
158 epoch_loss:tensor(0.3734, grad_fn=<AddBackward0>) total loss:tensor(87.664
9, grad_fn=<AddBackward0>)
159 epoch_loss:tensor(0.3724, grad_fn=<AddBackward0>) total loss:tensor(88.037
3, grad_fn=<AddBackward0>)
160 epoch_loss:tensor(0.3713, grad_fn=<AddBackward0>) total loss:tensor(88.408
6, grad_fn=<AddBackward0>)
161 epoch_loss:tensor(0.3703, grad_fn=<AddBackward0>) total loss:tensor(88.778
8, grad_fn=<AddBackward0>)
162 epoch_loss:tensor(0.3692, grad_fn=<AddBackward0>) total loss:tensor(89.148
1, grad_fn=<AddBackward0>)
163 epoch_loss:tensor(0.3683, grad_fn=<AddBackward0>) total loss:tensor(89.516
3, grad_fn=<AddBackward0>)
164 epoch_loss:tensor(0.3673, grad_fn=<AddBackward0>) total loss:tensor(89.883
6, grad_fn=<AddBackward0>)
165 epoch_loss:tensor(0.3664, grad_fn=<AddBackward0>) total loss:tensor(90.250
0, grad_fn=<AddBackward0>)
166 epoch_loss:tensor(0.3654, grad_fn=<AddBackward0>) total loss:tensor(90.615
4, grad_fn=<AddBackward0>)
167 epoch_loss:tensor(0.3645, grad_fn=<AddBackward0>) total loss:tensor(90.979
9, grad_fn=<AddBackward0>)
168 epoch_loss:tensor(0.3637, grad_fn=<AddBackward0>) total loss:tensor(91.343
6, grad_fn=<AddBackward0>)
169 epoch_loss:tensor(0.3628, grad_fn=<AddBackward0>) total loss:tensor(91.706
4, grad_fn=<AddBackward0>)
170 epoch_loss:tensor(0.3620, grad_fn=<AddBackward0>) total loss:tensor(92.068
4, grad_fn=<AddBackward0>)
171 epoch_loss:tensor(0.3612, grad_fn=<AddBackward0>) total loss:tensor(92.429
6, grad_fn=<AddBackward0>)
172 epoch_loss:tensor(0.3604, grad_fn=<AddBackward0>) total loss:tensor(92.789
9, grad_fn=<AddBackward0>)
173 epoch_loss:tensor(0.3596, grad_fn=<AddBackward0>) total loss:tensor(93.149
5, grad_fn=<AddBackward0>)
174 epoch_loss:tensor(0.3588, grad_fn=<AddBackward0>) total loss:tensor(93.508
3, grad_fn=<AddBackward0>)
175 epoch_loss:tensor(0.3581, grad_fn=<AddBackward0>) total loss:tensor(93.866
4, grad_fn=<AddBackward0>)
176 epoch_loss:tensor(0.3574, grad_fn=<AddBackward0>) total loss:tensor(94.223
8, grad_fn=<AddBackward0>)
177 epoch_loss:tensor(0.3567, grad_fn=<AddBackward0>) total loss:tensor(94.580
4, grad_fn=<AddBackward0>)
178 epoch_loss:tensor(0.3559, grad_fn=<AddBackward0>) total loss:tensor(94.936
4, grad_fn=<AddBackward0>)
179 epoch_loss:tensor(0.3553, grad_fn=<AddBackward0>) total loss:tensor(95.291
6, grad_fn=<AddBackward0>)
180 epoch_loss:tensor(0.3546, grad_fn=<AddBackward0>) total loss:tensor(95.646
2, grad_fn=<AddBackward0>)
181 epoch_loss:tensor(0.3540, grad_fn=<AddBackward0>) total loss:tensor(96.000
2, grad_fn=<AddBackward0>)
182 epoch_loss:tensor(0.3533, grad_fn=<AddBackward0>) total loss:tensor(96.353
5, grad_fn=<AddBackward0>)
183 epoch_loss:tensor(0.3527, grad_fn=<AddBackward0>) total loss:tensor(96.706
2, grad_fn=<AddBackward0>)
```

```
184 epoch_loss:tensor(0.3521, grad_fn=<AddBackward0>) total loss:tensor(97.058
3, grad_fn=<AddBackward0>)
185 epoch_loss:tensor(0.3515, grad_fn=<AddBackward0>) total loss:tensor(97.409
9, grad_fn=<AddBackward0>)
186 epoch_loss:tensor(0.3509, grad_fn=<AddBackward0>) total loss:tensor(97.760
8, grad_fn=<AddBackward0>)
187 epoch_loss:tensor(0.3504, grad_fn=<AddBackward0>) total loss:tensor(98.111
1, grad_fn=<AddBackward0>)
188 epoch_loss:tensor(0.3498, grad_fn=<AddBackward0>) total loss:tensor(98.461
0, grad_fn=<AddBackward0>)
189 epoch_loss:tensor(0.3493, grad_fn=<AddBackward0>) total loss:tensor(98.810
2, grad_fn=<AddBackward0>)
190 epoch_loss:tensor(0.3487, grad_fn=<AddBackward0>) total loss:tensor(99.159
0, grad_fn=<AddBackward0>)
191 epoch_loss:tensor(0.3482, grad_fn=<AddBackward0>) total loss:tensor(99.507
2, grad_fn=<AddBackward0>)
192 epoch_loss:tensor(0.3477, grad_fn=<AddBackward0>) total loss:tensor(99.854
9, grad_fn=<AddBackward0>)
193 epoch_loss:tensor(0.3472, grad_fn=<AddBackward0>) total loss:tensor(100.20
21, grad_fn=<AddBackward0>)
194 epoch_loss:tensor(0.3467, grad_fn=<AddBackward0>) total loss:tensor(100.54
88, grad_fn=<AddBackward0>)
195 epoch_loss:tensor(0.3462, grad_fn=<AddBackward0>) total loss:tensor(100.89
50, grad_fn=<AddBackward0>)
196 epoch_loss:tensor(0.3458, grad_fn=<AddBackward0>) total loss:tensor(101.24
08, grad_fn=<AddBackward0>)
197 epoch_loss:tensor(0.3453, grad_fn=<AddBackward0>) total loss:tensor(101.58
61, grad_fn=<AddBackward0>)
198 epoch_loss:tensor(0.3449, grad_fn=<AddBackward0>) total loss:tensor(101.93
10, grad_fn=<AddBackward0>)
199 epoch_loss:tensor(0.3444, grad_fn=<AddBackward0>) total loss:tensor(102.27
54, grad_fn=<AddBackward0>)
200 epoch_loss:tensor(0.3440, grad_fn=<AddBackward0>) total loss:tensor(102.61
94, grad_fn=<AddBackward0>)
201 epoch_loss:tensor(0.3436, grad_fn=<AddBackward0>) total loss:tensor(102.96
30, grad_fn=<AddBackward0>)
202 epoch_loss:tensor(0.3432, grad_fn=<AddBackward0>) total loss:tensor(103.30
62, grad_fn=<AddBackward0>)
203 epoch_loss:tensor(0.3428, grad_fn=<AddBackward0>) total loss:tensor(103.64
90, grad_fn=<AddBackward0>)
204 epoch_loss:tensor(0.3424, grad_fn=<AddBackward0>) total loss:tensor(103.99
14, grad_fn=<AddBackward0>)
205 epoch_loss:tensor(0.3420, grad_fn=<AddBackward0>) total loss:tensor(104.33
33, grad_fn=<AddBackward0>)
206 epoch_loss:tensor(0.3416, grad_fn=<AddBackward0>) total loss:tensor(104.67
50, grad_fn=<AddBackward0>)
207 epoch_loss:tensor(0.3412, grad_fn=<AddBackward0>) total loss:tensor(105.01
62, grad_fn=<AddBackward0>)
208 epoch_loss:tensor(0.3409, grad_fn=<AddBackward0>) total loss:tensor(105.35
71, grad_fn=<AddBackward0>)
209 epoch_loss:tensor(0.3405, grad_fn=<AddBackward0>) total loss:tensor(105.69
76, grad_fn=<AddBackward0>)
210 epoch_loss:tensor(0.3402, grad_fn=<AddBackward0>) total loss:tensor(106.03
77, grad_fn=<AddBackward0>)
211 epoch_loss:tensor(0.3398, grad_fn=<AddBackward0>) total loss:tensor(106.37
75, grad_fn=<AddBackward0>)
212 epoch_loss:tensor(0.3395, grad_fn=<AddBackward0>) total loss:tensor(106.71
70, grad_fn=<AddBackward0>)
213 epoch_loss:tensor(0.3391, grad_fn=<AddBackward0>) total loss:tensor(107.05
61, grad_fn=<AddBackward0>)
214 epoch_loss:tensor(0.3388, grad_fn=<AddBackward0>) total loss:tensor(107.39
49, grad_fn=<AddBackward0>)
215 epoch_loss:tensor(0.3385, grad_fn=<AddBackward0>) total loss:tensor(107.73
34, grad_fn=<AddBackward0>)
216 epoch_loss:tensor(0.3382, grad_fn=<AddBackward0>) total loss:tensor(108.07
16, grad_fn=<AddBackward0>)
217 epoch_loss:tensor(0.3379, grad_fn=<AddBackward0>) total loss:tensor(108.40
95, grad_fn=<AddBackward0>)
218 epoch_loss:tensor(0.3376, grad_fn=<AddBackward0>) total loss:tensor(108.74
```

```
71, grad_fn=<AddBackward0>)
219 epoch_loss:tensor(0.3373, grad_fn=<AddBackward0>) total loss:tensor(109.08
43, grad_fn=<AddBackward0>)
220 epoch_loss:tensor(0.3370, grad_fn=<AddBackward0>) total loss:tensor(109.42
13, grad_fn=<AddBackward0>)
221 epoch_loss:tensor(0.3367, grad_fn=<AddBackward0>) total loss:tensor(109.75
80, grad_fn=<AddBackward0>)
222 epoch_loss:tensor(0.3364, grad_fn=<AddBackward0>) total loss:tensor(110.09
44, grad_fn=<AddBackward0>)
223 epoch_loss:tensor(0.3361, grad_fn=<AddBackward0>) total loss:tensor(110.43
06, grad_fn=<AddBackward0>)
224 epoch_loss:tensor(0.3359, grad_fn=<AddBackward0>) total loss:tensor(110.76
64, grad_fn=<AddBackward0>)
225 epoch_loss:tensor(0.3356, grad_fn=<AddBackward0>) total loss:tensor(111.10
20, grad_fn=<AddBackward0>)
226 epoch_loss:tensor(0.3353, grad_fn=<AddBackward0>) total loss:tensor(111.43
74, grad_fn=<AddBackward0>)
227 epoch_loss:tensor(0.3351, grad_fn=<AddBackward0>) total loss:tensor(111.77
25, grad_fn=<AddBackward0>)
228 epoch_loss:tensor(0.3348, grad_fn=<AddBackward0>) total loss:tensor(112.10
73, grad_fn=<AddBackward0>)
229 epoch_loss:tensor(0.3346, grad_fn=<AddBackward0>) total loss:tensor(112.44
19, grad_fn=<AddBackward0>)
230 epoch_loss:tensor(0.3343, grad_fn=<AddBackward0>) total loss:tensor(112.77
62, grad_fn=<AddBackward0>)
231 epoch_loss:tensor(0.3341, grad_fn=<AddBackward0>) total loss:tensor(113.11
03, grad_fn=<AddBackward0>)
232 epoch_loss:tensor(0.3339, grad_fn=<AddBackward0>) total loss:tensor(113.44
42, grad_fn=<AddBackward0>)
233 epoch_loss:tensor(0.3336, grad_fn=<AddBackward0>) total loss:tensor(113.77
78, grad_fn=<AddBackward0>)
234 epoch_loss:tensor(0.3334, grad_fn=<AddBackward0>) total loss:tensor(114.11
12, grad_fn=<AddBackward0>)
235 epoch_loss:tensor(0.3332, grad_fn=<AddBackward0>) total loss:tensor(114.44
44, grad_fn=<AddBackward0>)
236 epoch_loss:tensor(0.3330, grad_fn=<AddBackward0>) total loss:tensor(114.77
74, grad_fn=<AddBackward0>)
237 epoch_loss:tensor(0.3328, grad_fn=<AddBackward0>) total loss:tensor(115.11
02, grad_fn=<AddBackward0>)
238 epoch_loss:tensor(0.3325, grad_fn=<AddBackward0>) total loss:tensor(115.44
27, grad_fn=<AddBackward0>)
239 epoch_loss:tensor(0.3323, grad_fn=<AddBackward0>) total loss:tensor(115.77
50, grad_fn=<AddBackward0>)
240 epoch_loss:tensor(0.3321, grad_fn=<AddBackward0>) total loss:tensor(116.10
72, grad_fn=<AddBackward0>)
241 epoch_loss:tensor(0.3319, grad_fn=<AddBackward0>) total loss:tensor(116.43
91, grad_fn=<AddBackward0>)
242 epoch_loss:tensor(0.3317, grad_fn=<AddBackward0>) total loss:tensor(116.77
08, grad_fn=<AddBackward0>)
243 epoch_loss:tensor(0.3315, grad_fn=<AddBackward0>) total loss:tensor(117.10
23, grad_fn=<AddBackward0>)
244 epoch_loss:tensor(0.3313, grad_fn=<AddBackward0>) total loss:tensor(117.43
37, grad_fn=<AddBackward0>)
245 epoch_loss:tensor(0.3311, grad_fn=<AddBackward0>) total loss:tensor(117.76
48, grad_fn=<AddBackward0>)
246 epoch_loss:tensor(0.3310, grad_fn=<AddBackward0>) total loss:tensor(118.09
58, grad_fn=<AddBackward0>)
247 epoch_loss:tensor(0.3308, grad_fn=<AddBackward0>) total loss:tensor(118.42
66, grad_fn=<AddBackward0>)
248 epoch_loss:tensor(0.3306, grad_fn=<AddBackward0>) total loss:tensor(118.75
72, grad_fn=<AddBackward0>)
249 epoch_loss:tensor(0.3304, grad_fn=<AddBackward0>) total loss:tensor(119.08
76, grad_fn=<AddBackward0>)
validation loss is equal to: tensor(0.5815, grad_fn=<NllLossBackward>)
              precision    recall  f1-score   support

           0       0.88      0.82      0.85        17
           1       0.81      0.87      0.84        15
```

accuracy			0.84	32
macro avg	0.84	0.85	0.84	32
weighted avg	0.85	0.84	0.84	32