Social network

首先 通过G = nx.Graph() 去建立一个新的神经网络

1. add\_nodes() 括号里面输入一个字符 字符串使用引号注释 G.nodes 来查看Nodeview

多量添加使用 G.add\_notes\_from(range())等输入list或者array

使用 nx.draw(G, node\_color=’’) 来画出这个神经网路的图 大小 label 都可以在此函数中改变

使用G.add\_edge(1，2) 括号中添加连线 此为添加1与2点的连线 可使用G.add\_edges\_from()添加多个 如e = [(1,2)(2,3)(4,5)] 在其中添加e 即可解决

使用 list(nx.connected\_components(G))即可显示那些点相连

使用nx.adjacency\_matrix(G) 来显示邻接矩阵

使用nx.adjacency\_matrix(G).A 函数从numpy获得一个完整的密集数组

使用print(nx.adjacency\_matrix(G)) 来显示所有相连节点

使用 G.adj 也可以查看类似类型 如 G.adj[1] 可以查看相连接的

使用plt.imshow(adj) 创建2维numpy热图

以上都是在无权值情况下

使用此方法 循环加入权值 权值在0-1之间

for e in G.edges():  
 G[e[0]][e[1]]['weight'] = random.uniform(0, 1)

print(np.around(nx.adjacency\_matrix(G),4)) 这个4 是为了简化 将权值四舍五入到小数点4位

G.degree 显示所有链接点的权值

Collections.Counter(degree\_sequence)显示每个度值在网络之中出现的次数

在一个图里分开显示点和线

nx.draw\_networkx\_nodes(G, pos, node\_color='darkblue', node\_size=20) #<-- Plot the nodes  
nx.draw\_networkx\_edges(G, pos, alpha=0.4) #<-- Plot the edges