

product : [t, test\_size, 2] tensor  
 y\_all : [test\_size, 2] tensor  
 答案  
 PandY0 PandY1 PandY2 PandY3 PandY4

N: 5

{d: I (ac\_size, dis\_size)} tensor  
 (ei: list->tensor [2, test\_size] 预测集输出)

pred = predall(dijsoftmax, 预测矩阵, I (test\_size))

trh: 真实值矩阵 [dis\_size, dis\_size] 正确识别  
 tth: 预测值矩阵 [dis\_size, dis\_size] 非预测识别

Rh: list->tensor strtrh

(labelh: list->tensor strtrh

def calculate\_AUC\_AUPR (Rh, labelh)

R: 矩阵 Rh 和标签h [dis\_size, hc\_size]

(label: 矩阵 labelh 和标签h [dis\_size, hc\_size])

f: [dis\_size, 1] 预测列 (8个)

def calculate\_TPR\_FPR (RD, f, B)

old\_id: RD 的每行从大到小排的 F 标 [dis\_size, hc\_size]

min\_f: f 的最小值

max\_f: f 的最大值

TP, FN: 按照矩阵每行排序后的行数 [dis\_size, 1]

FP, TN: 按照矩阵每行等于0的个数 [dis\_size, 1]

TP: [dis\_size, max\_f]

TPs: [dis\_size, max\_f]

PP: [dis\_size, max\_f]

Fp2: [dis\_size, max\_f]

P: [dis\_size, max\_f]

P2: [dis\_size, max\_f]

for i in range (RD.shape[0]):

for j in range (int(f[i])):

if j == 0:

if RD[i][j] == 1:

TPP, FP2, TPs = 1, 0,

TPC, TN, TPs = 1, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,

PE, FPs = TP, [j] / hc

else:

TPP, FP2, TPs = 0, 0,

TPC, TN, TPs = 0, 0,