HR@1 0.442443 NDCG@1 0.442443

HR@2 0.624932 NDCG@2 0.557581

HR@3 0.716267 NDCG@3 0.603248

HR@4 0.768150 NDCG@4 0.625593

HR@5 0.804720 NDCG@5 0.639740

HR@6 0.832823 NDCG@6 0.649751

HR@7 0.852819 NDCG@7 0.656416

HR@8 0.867051 NDCG@8 0.660906

HR@9 0.880922 NDCG@9 0.665081

HR@10 0.890830 NDCG@10 0.667946

weighted\_output = tf.reduce\_sum(neighborhood, axis=2, #连加 得oui 特定用户、项目和邻域之间的关系组成邻域的加权和  
 name='OutputNeighborhood') \* query #改

HR@1 0.234192 NDCG@1 0.234192

HR@2 0.341920 NDCG@2 0.302161

HR@3 0.392362 NDCG@3 0.327382

HR@4 0.427671 NDCG@4 0.342588

HR@5 0.457035 NDCG@5 0.353948

HR@6 0.478292 NDCG@6 0.361520

HR@7 0.495226 NDCG@7 0.367165

HR@8 0.511439 NDCG@8 0.372279

HR@9 0.527112 NDCG@9 0.376997

HR@10 0.539002 NDCG@10 0.380434

neighborhood = tf.matmul(c\_temp, probs\_temp, name='WeightedNeighborhood') #改tf.matmul

probs\_temp = tf.transpose(tf.expand\_dims(attention, 1, name='TransformAttention'), [0, 2, 1]) #+ query\_expanded #puiv 改  
 #tf.transpose [0,2,1]

HR@1 0.441902 NDCG@1 0.441902

HR@2 0.618988 NDCG@2 0.553631

HR@3 0.710863 NDCG@3 0.599568

HR@4 0.763466 NDCG@4 0.622223

HR@5 0.800036 NDCG@5 0.636370

HR@6 0.823095 NDCG@6 0.644584

HR@7 0.843812 NDCG@7 0.651490

HR@8 0.861466 NDCG@8 0.657059

HR@9 0.876239 NDCG@9 0.661506

HR@10 0.886687 NDCG@10 0.664526

Hop=1

HR@1 0.292200 NDCG@1 0.292200

HR@2 0.422807 NDCG@2 0.374604

HR@3 0.490002 NDCG@3 0.408201

HR@4 0.542245 NDCG@4 0.430701

HR@5 0.579896 NDCG@5 0.445266

HR@6 0.613223 NDCG@6 0.457138

HR@7 0.638804 NDCG@7 0.465665

HR@8 0.664565 NDCG@8 0.473791

HR@9 0.687984 NDCG@9 0.480841

HR@10 0.708161 NDCG@10 0.486674

Hop=3

HR@1 0.443344 NDCG@1 0.443344

HR@2 0.616646 NDCG@2 0.552685

HR@3 0.707260 NDCG@3 0.597992

HR@4 0.765988 NDCG@4 0.623285

HR@5 0.800937 NDCG@5 0.636805

HR@6 0.827238 NDCG@6 0.646174

HR@7 0.850297 NDCG@7 0.653860

HR@8 0.867231 NDCG@8 0.659202

HR@9 0.881643 NDCG@9 0.663541

HR@10 0.890290 NDCG@10 0.666040

HR@1 0.435417 NDCG@1 0.435417

HR@2 0.611241 NDCG@2 0.546350

HR@3 0.699694 NDCG@3 0.590576

HR@4 0.761484 NDCG@4 0.617188

HR@5 0.798775 NDCG@5 0.631614

HR@6 0.828860 NDCG@6 0.642330

HR@7 0.848496 NDCG@7 0.648876

HR@8 0.867051 NDCG@8 0.654729

HR@9 0.878220 NDCG@9 0.658091

HR@10 0.888849 NDCG@10 0.661164

HR@1 0.437579 NDCG@1 0.437579

HR@2 0.612502 NDCG@2 0.547943

HR@3 0.705458 NDCG@3 0.594421

HR@4 0.760223 NDCG@4 0.618007

HR@5 0.795532 NDCG@5 0.631667

HR@6 0.823275 NDCG@6 0.641549

HR@7 0.845794 NDCG@7 0.649055

HR@8 0.861106 NDCG@8 0.653886

HR@9 0.876779 NDCG@9 0.658604

HR@10 0.888128 NDCG@10 0.661884

将嵌入矩阵更改为特定项目的另一个嵌入矩阵

HR@1 0.443344 NDCG@1 0.443344

HR@2 0.615385 NDCG@2 0.551889

HR@3 0.705819 NDCG@3 0.597106

HR@4 0.762926 NDCG@4 0.621701

HR@5 0.803099 NDCG@5 0.637242

HR@6 0.831202 NDCG@6 0.647253

HR@7 0.850477 NDCG@7 0.653678

HR@8 0.866150 NDCG@8 0.658622

HR@9 0.881463 NDCG@9 0.663232

HR@10 0.893172 NDCG@10 0.666616

嵌入矩阵均为项目，项目级和用户级同时加入

HR@1 0.436138 NDCG@1 0.436138

HR@2 0.608179 NDCG@2 0.544683

HR@3 0.699514 NDCG@3 0.590351

HR@4 0.756080 NDCG@4 0.614713

HR@5 0.797694 NDCG@5 0.630811

HR@6 0.826698 NDCG@6 0.641143

HR@7 0.846694 NDCG@7 0.647808

HR@8 0.863448 NDCG@8 0.653093

HR@9 0.875878 NDCG@9 0.656835

HR@10 0.887948 NDCG@10 0.660324

同时加入用户和项目的权重矩阵

AttentionOutput = namedtuple("AttentionOutput", ['weight', 'output']) #加output\_u'output\_u'

probs\_temp\_u = tf.expand\_dims(attention, 1, name='UserTransformAttention') #加

c\_temp = tf.transpose(output\_memory, [0, 2, 1], #转置  
 name='TransformOutputMemory')

neighborhood\_u = tf.multiply(c\_temp, probs\_temp\_u, name='UserWeightedNeighborhood') #加

return AttentionOutput(weight=attention, output=weighted\_output) #, output\_u=user\_weighted\_output

query = self.\_activation\_fn(hop\_mapping(query) + memory\_hop.output) #加memory\_hop.output\_u + memory\_hop.output\_u

cmn

#neighbor\_add = layer.output\_u #加

self.score = self.\_output\_module(tf.concat([self.\_cur\_user \* self.\_cur\_item, #concat()用来拼接张量 axis=1代表在第1个维度拼接  
 neighbor\_ori], axis=1)) #mu\*ei #改, neighbor\_add

#neighbor\_negative\_add = neg\_layer.output\_u #改

negative\_output = self.\_output\_module(tf.concat(  
 [self.\_cur\_user \* self.\_cur\_item\_negative, neighbor\_negative\_ori], axis=1)) #改, neighbor\_negative\_add

HR@1 0.444605 NDCG@1 0.444605

HR@2 0.617727 NDCG@2 0.553832

HR@3 0.704738 NDCG@3 0.597338

HR@4 0.760764 NDCG@4 0.621467

HR@5 0.798235 NDCG@5 0.635963

HR@6 0.826878 NDCG@6 0.646166

HR@7 0.851558 NDCG@7 0.654393

HR@8 0.866691 NDCG@8 0.659166

HR@9 0.881103 NDCG@9 0.663505

HR@10 0.891911 NDCG@10 0.666629.

Item-hop=1

HR@1 0.274905 NDCG@1 0.274905

HR@2 0.387137 NDCG@2 0.345716

HR@3 0.442443 NDCG@3 0.373369

HR@4 0.479914 NDCG@4 0.389506

HR@5 0.507836 NDCG@5 0.400308

HR@6 0.532517 NDCG@6 0.409100

HR@7 0.550712 NDCG@7 0.415165

HR@8 0.568907 NDCG@8 0.420905

HR@9 0.586741 NDCG@9 0.426273

HR@10 0.602414 NDCG@10 0.430804

Item-hop=4

HR@1 0.452711 NDCG@1 0.452711

HR@2 0.626374 NDCG@2 0.562280

HR@3 0.713385 NDCG@3 0.605786

HR@4 0.769411 NDCG@4 0.629915

HR@5 0.804360 NDCG@5 0.643435

HR@6 0.834264 NDCG@6 0.654087

HR@7 0.856062 NDCG@7 0.661353

HR@8 0.868852 NDCG@8 0.665388

HR@9 0.882364 NDCG@9 0.669455

HR@10 0.893353 NDCG@10 0.672632

得分函数改为用sigmoid

HR@1 0.447667 NDCG@1 0.447667

HR@2 0.626374 NDCG@2 0.560418

HR@3 0.714286 NDCG@3 0.604374

HR@4 0.765628 NDCG@4 0.626486

HR@5 0.803819 NDCG@5 0.641261

HR@6 0.830301 NDCG@6 0.650694

HR@7 0.853180 NDCG@7 0.658320

HR@8 0.869213 NDCG@8 0.663378

HR@9 0.882003 NDCG@9 0.667228

HR@10 0.892452 NDCG@10 0.670248

Sigmoid+1

HR@1 0.445145 NDCG@1 0.445145

HR@2 0.624752 NDCG@2 0.558465

HR@3 0.714646 NDCG@3 0.603411

HR@4 0.767249 NDCG@4 0.626066

HR@5 0.803639 NDCG@5 0.640144

HR@6 0.832102 NDCG@6 0.650283

HR@7 0.855522 NDCG@7 0.658089

HR@8 0.872636 NDCG@8 0.663488

HR@9 0.884525 NDCG@9 0.667067

HR@10 0.897316 NDCG@10 0.670764

Sigmoid+0.6

HR@1 0.440281 NDCG@1 0.440281

HR@2 0.616646 NDCG@2 0.551555

HR@3 0.709962 NDCG@3 0.598213

HR@4 0.765628 NDCG@4 0.622187

HR@5 0.802198 NDCG@5 0.636334

HR@6 0.828680 NDCG@6 0.645767

HR@7 0.851919 NDCG@7 0.653513

HR@8 0.870294 NDCG@8 0.659310

HR@9 0.883625 NDCG@9 0.663323

HR@10 0.895154 NDCG@10 0.666656

tanh

HR@1 0.443344 NDCG@1 0.443344

HR@2 0.620969 NDCG@2 0.555413

HR@3 0.712124 NDCG@3 0.600990

HR@4 0.764006 NDCG@4 0.623335

HR@5 0.802738 NDCG@5 0.638318

HR@6 0.829580 NDCG@6 0.647880

HR@7 0.852279 NDCG@7 0.655446

HR@8 0.867952 NDCG@8 0.660390

HR@9 0.882364 NDCG@9 0.664728

HR@10 0.892992 NDCG@10 0.667801

+1.2

HR@1 0.441182 NDCG@1 0.441182

HR@2 0.615925 NDCG@2 0.551433

HR@3 0.708521 NDCG@3 0.597730

HR@4 0.765087 NDCG@4 0.622092

HR@5 0.798775 NDCG@5 0.635124

HR@6 0.828319 NDCG@6 0.645648

HR@7 0.850477 NDCG@7 0.653034

HR@8 0.867591 NDCG@8 0.658433

HR@9 0.878761 NDCG@9 0.661795

HR@10 0.889930 NDCG@10 0.665024