1. PITF，代码修改后：100个负例样本采样，

Precisions: 0.13201219512195206

Recall: 0.5179442508710802

F1: 0.21039858264293398

大体上结果趋于稳定（还是稍差于远远0.226，不同的内容仅在于初始化参数以及每次抽样的内容不同）

1. ATPITF 100个负例样本， gamma =0.2， 10个序列记忆：

Precisions: 0.09115853658536624

Recall: 0.3645143728222997

F1: 0.14584407413645742

Gamma = 0.1, 10 个记忆序列， 10个负样本：

Precisions: 0.11250000000000059

Recall: 0.44815258420441345

F1: 0.1798517197402756

Gamma = 0.2, 10个记忆序列，10个负样本：

Precisions: 0.0966463414634151

Recall: 0.38755807200929154

F1: 0.15471180651030605

Gamma = 0.3, 10个记忆序列， 10个负样本

Precisions: 0.08993902439024429

Recall: 0.35981416957026713

F1: 0.1439070841852747

500轮迭代：

Gmama=0.3, 5 个记忆序列：

Precisions: 0.10731707317073227

Recall: 0.42207099303135887

F1: 0.1711237050254876

Gmama=0.2, 5 个记忆序列

Precisions: 0.1134146341463421

Recall: 0.4483304297328687

F1: 0.18103312315267292

Gmama=0.1, 5 个记忆序列

Precisions: 0.11859756097561044

Recall: 0.46718205574912886

F1: 0.18917234660097162

500轮迭代， Gmma=0.1,0.2,0,3, 2个记忆序列，10个负样本：

Precisions: 0.12500000000000078

Recall: 0.48865055168408816

F1: 0.19907525151857583

Precisions: 0.12164634146341537

Recall: 0.4764046167247387

F1: 0.1938058216858919

Precisions: 0.12439024390243981

Recall: 0.4901749419279906

F1: 0.19842640613915466

Gama = 0.4, 0.5, 0.8:

Precisions: 0.12286585365853733

Recall: 0.48723867595818826

F1: 0.19624504638464527

Precisions: 0.12865853658536666

Recall: 0.5073243321718931

F1: 0.20526215204169884

Precisions: 0.13018292682926916

Recall: 0.512848432055749

F1: 0.20765428927318372

Gama = 1:

Precisions: 0.1292682926829277

Recall: 0.508961236933798

F1: 0.20617206533747612

Gamma = 0.2 0.4,,0.5，0.6 记忆长度为1：

0．2：

Precisions: 0.12591463414634227

Recall: 0.49355400696864105

F1: 0.20064186657475197

0.4：

Precisions: 0.13231707317073257

Recall: 0.5167247386759583

F1: 0.21068443914878124

0.5

Precisions: 0.13323170731707407

Recall: 0.5239909988385599

F1: 0.21244614569816275

0.6：

Precisions: 0.1347560975609765

Recall: 0.5264046167247388

F1: 0.21458090402283891

0.7：

Precisions: 0.13414634146341547

Recall: 0.5230255516840883

F1: 0.21352697819825034

0.8：

Precisions: 0.13841463414634234

Recall: 0.5392857142857143

F1: 0.22028920308483402

0.9：

Precisions: 0.13689024390243992

Recall: 0.5415577816492451

F1: 0.21853988522391785

NeuralPITF 100个负例样本：

Precisions: 0.13414634146341547

Recall: 0.5194178281068526

F1: 0.21322466737190385

500轮迭代最好结果：0.2188，可以尝试调整某些参数试试

Precisions: 0.13109756097561057

Recall: 0.507146486643438

F1: 0.20833932632612148

best result: 0.21878106231079425

1. TransPITF, 100个负例样本， 500轮迭代，使用单层感知机作为投影， Relu为激活函数：

Precisions: 0.06951219512195134

Recall: 0.2777221254355401

F1: 0.11119335520730575

best result: 0.11450786644748633

使用 sigmoid函数为激活函数，100轮迭代：

Precisions: 0.045731707317073066

Recall: 0.18823678861788617

F1: 0.07358588761259517

best result: 0.07574009388190035

500轮迭代：

Precisions: 0.0652439024390245

Recall: 0.2623657084785135

F1: 0.10450097992776139

best result: 0.10821259771248833

1. TimeAttentionPITF pytorch实现， 10个记忆序列，gamma = 0.5, 10个负例，200次迭代

Precisions: 0.12987804878048856

Recall: 0.49223286875725913

F1: 0.2055268368954031

best result: 0.20586795283799741

5个记忆序列，gamma=0.5,100个负例，200次迭代：

Precisions: 0.13292682926829347

Recall: 0.5049361207897795

F1: 0.21045134386158235

best result: 0.21352673759319257

10个记忆序列，仅使用时间，gamma=1 200轮迭代：

Precisions: 0.09420731707317101

Recall: 0.3378847270615563

F1: 0.14733533768349055

best result: 0.1508455652986541

1. AttentionPITF pytorch实现： m=5 gamma=0.5, 100负例，100轮迭代：

Precisions: 0.131402439024391

Recall: 0.5061665214866434

F1: 0.20864100856643147

best result: 0.21226142707232684

仅使用序列作为user embedding:

10个tag，200轮迭代：

在attention机制上，加一层多层感知机：m=5,gamma=0.5, 100负例，200轮迭代

Precisions: 0.13048780487804962

Recall: 0.5090011614401858

F1: 0.20772350340648446

best result: 0.21047880973912145

多层感知机是对于所有的tag history而言， m=5, gamma=0.05, 100负例，100轮迭代：

Precisions: 0.10274390243902486

Recall: 0.41072517421602783

F1: 0.16437019928759053

best result: 0.1647451660003359 (Adam优化)

Precisions: 0.1271341463414642

Recall: 0.4940839140534264

F1: 0.20223152106782621

best result: 0.20234195229453708（SGD优化）

我们尝试，直接使用历史记录的行为作为user embedding；

10个用户序列，200轮迭代：

Precisions: 0.12317073170731785

Recall: 0.48275261324041824

F1: 0.19626572602701398

best result: 0.20024086198902735

1. RNNAttention实现： m=5, gamma =0.5, dropout=0.2, 100负例，100轮迭代

Precisions: 0.11737804878048844

Recall: 0.46027874564459936

F1: 0.18705439485971506

best result: 0.18844768344301122

m=5, gamma=0,5, dropout=0.5:

Precisions: 0.08780487804878079

Recall: 0.3542211091753776

F1: 0.1407263020383252

best result: 0.15346059781670346（adam优化）

Precisions: 0.12073170731707392

Recall: 0.4734139082462255

F1: 0.19239751304409333

best result: 0.19426060632475461（SGD优化）

m=5, gamma=0.5, dropout = 0.2, tagMLP:

Precisions: 0.09329268292682968

Recall: 0.3736933797909409

F1: 0.1493100577340418

best result: 0.16231607820242921（adam优化）

Precisions: 0.11920731707317136

Recall: 0.4626923635307782

F1: 0.18957327912434044

best result: 0.19394284533268194（SGD优化）

m=5, gamma=0.5, dropout = 0.5, tagMLP

Precisions: 0.11920731707317136

Recall: 0.4626923635307782

F1: 0.18957327912434044

best result: 0.19394284533268194

m=5, gamma=0.5, dropout=0.5, rnn=GRU (tag MLP)

Precisions: 0.12164634146341528

Recall: 0.4691855400696864

F1: 0.193201166697083

best result: 0.19452672359582274

m=5, gamma=0.5, dropout=0.5, (tag MLP) 500 轮迭代

Precisions: 0.11890243902439093

Recall: 0.46426756678281084

F1: 0.18931887957435753

best result: 0.1948737963898516m=5, gamma=0.5, dropout=0.5, rnn=GRU (tag MLP) 500 轮迭代

Precisions: 0.12012195121951293

Recall: 0.47227061556329863

F1: 0.19152862823109257

best result: 0.19786425121807558

**修正bug: embedding自动初始化（忽略0元素）**

m=5, gamma=0.5, dropout=0.5, (tag MLP) 200轮迭代，RNN

Precisions: 0.1277439024390252

Recall: 0.49987659698025566

F1: 0.2034866206418751

best result: 0.21887525320001835

m=5, gamma=0.5, (tag MLP) sotf attention机制 100轮迭代：

Precisions: 0.14237804878048868

Recall: 0.5609030197444831

F1: 0.2271077129199527

best result: 0.2279936784723507

m=5, gamma=0.5, dropout=0.5, **(user MLP)** sotf attention机制, LSTM 100轮迭代：

Precisions: 0.12439024390243977

Recall: 0.49327090592334494

F1: 0.19867880087678738

best result: 0.19867880087678738

m=5, gamma=0.5, dropout=0.5, (tag MLP) sotf attention机制, GRU 100轮迭代：

Precisions: 0.1320121951219521

Recall: 0.5249165214866434

F1: 0.2109677366974292

best result: 0.2119925511293255

根据之前错误代码的最好参数，m=10, gamma=0.2, tag MLP, 直接 soft attention机制，100轮迭代：

Precisions: 0.13993902439024486

Recall: 0.5522793263646921

F1: 0.22329783669585979

best result: 0.22343162927507346

m=10, gamma=0.5,tagMLP, 直接sotfattention机制：

Precisions: 0.14176829268292776

Recall: 0.5554043263646922

F1: 0.2258801362709737

best result: 0.22594310126907516 （SGD优化）

**Precisions: 0.14176829268292776**

**Recall: 0.5554043263646922**

**F1: 0.2258801362709737**

**best result: 0.22594310126907516** (adam优化)

Precisions: 0.11646341463414704

Recall: 0.46390461672473865

F1: 0.18618501643450697

best result: 0.19195965307743026（adam, 速率0.001）优化

Precisions: 0.09420731707317115

Recall: 0.36858667247386767

F1: 0.15006055526640066

best result: 0.16193725567252915 （SGD, 速率0.001优化）

**m=8, gamma=0.4,tagMLP,直接attention（调参中最好的结果）：**

Precisions: 0.09420731707317115

Recall: 0.36858667247386767

F1: 0.15006055526640066

best result: 0.16193725567252915 （结果并不理想，还需要进一步调整）

m=5, gamma=0.5, tagMLP, tag query attention：

Precisions: 0.09207317073170773

Recall: 0.3682346109175378

F1: 0.14731242682388399

best result: 0.1619087825119503 (这个结果很不理想，需要找原因)

**m=5, gamma=0.5 tagMLP, time attention:**

Precisions: 0.14481707317073267

Recall: 0.5599920150987224

F1: 0.23012303892021838

best result: 0.23012303892021838

**我们调整调参策略，将gamma=0.5, 调整基于序列的长度，( user query attention ,tag mlp):**

gamma: 0.500000, the length: 2, best\_result: 0.227746

gamma: 0.500000, the length: 4, best\_result: 0.221388

gamma: 0.500000, the length: 5, best\_result: 0.218436

gamma: 0.500000, the length: 6, best\_result: 0.224041

gamma: 0.500000, the length: 8, best\_result: 0.226556

gamma: 0.500000, the length: 10, best\_result: 0.225943

gamma: 0.500000, the length: 15, best\_result: 0.225072

gamma: 0.500000, the length: 20, best\_result: 0.215459

**将gamma=0.5, 调整基于序列的长度，( user query attention ,tag mlp, GRU, dropout=0.5):**

gamma: 0.500000, the length: 2, best\_result: 0.213813

gamma: 0.500000, the length: 4, best\_result: 0.207712

gamma: 0.500000, the length: 5, best\_result: 0.202583

gamma: 0.500000, the length: 6, best\_result: 0.206689

gamma: 0.500000, the length: 8, best\_result: 0.203824

gamma: 0.500000, the length: 10, best\_result: 0.209146

gamma: 0.500000, the length: 15, best\_result: 0.207551

gamma: 0.500000, the length: 20, best\_result: 0.203163

**固定length值，调整gamma值（user query attention, tag mlp）:**

**Length=2:**

gamma: 0.200000, the length: 2, best\_result: 0.223821

gamma: 0.400000, the length: 2, best\_result: 0.222328

gamma: 0.500000, the length: 2, best\_result: 0.220745

gamma: 0.600000, the length: 2, best\_result: 0.222909

gamma: 0.800000, the length: 2, best\_result: 0.216810

gamma: 1.000000, the length: 2, best\_result: 0.117546

**Length=8:**

gamma: 0.200000, the length: 8, best\_result: 0.222324

gamma: 0.400000, the length: 8, best\_result: 0.222611

gamma: 0.500000, the length: 8, best\_result: 0.217583

gamma: 0.600000, the length: 8, best\_result: 0.226306

gamma: 0.800000, the length: 8, best\_result: 0.222169

gamma: 1.000000, the length: 8, best\_result: 0.214407

**TimeAttention, 固定length =5, 调整 gamma (没有进行 tag MLP):**

gamma: 0.200000, the length: 5, best\_result: 0.223850

gamma: 0.400000, the length: 5, best\_result: 0.225248

gamma: 0.500000, the length: 5, best\_result: 0.224031

gamma: 0.600000, the length: 5, best\_result: 0.230155

gamma: 0.800000, the length: 5, best\_result: 0.227020

gamma: 1.000000, the length: 5, best\_result: 0.216943

**我们复现每种结果中最好的一些情况(发现和第一次复现有些不一样，说明我们随机的种子还有些问题)：**

1. Softattention model, gamma=0.5, m=8

Precisions: 0.14390243902439123

Recall: 0.5650587979094076

F1: 0.22938726400057663

best result: 0.22938726400057663

1. Sotfattention model, gamma=0.5, m=2

Precisions: 0.14176829268292776

Recall: 0.5598105400696866

F1: 0.22624224331343737

best result: 0.22774627010764623

1. Softattention model, gamma=0.6, m=8

Precisions: 0.14024390243902532

Recall: 0.5487224157955866

F1: 0.22339255464368651

best result: 0.2257157811202952

1. TimeAttention gamma=0.6, m=5

Precisions: 0.14481707317073267

Recall: 0.5602460801393728

F1: 0.23014448337076884

best result: 0.23015519980155918

1. RNN attention model gamma=0.6, m=2

Precisions: 0.13109756097561065

Recall: 0.5229711091753776

F1: 0.20964232045474684

best result: 0.21015461047047762

1. RNN attention model gamma=0.6, m=10

Precisions: 0.12652439024390322

Recall: 0.509349593495935

F1: 0.202697856449558

best result: 0.20279824508509164

**使用最好结果，跑一个稀疏数据集：**

**其他数据集：**

Movielens core1:

softattention model gamma=0.5, m=8:

Precisions: 0.08927038626609511

Recall: 0.3419047619047618

F1: 0.14157573919061203

best result: 0.1445964836569277

TimeAttention gamma=0.6, m=5:

Precisions: 0.09236051502145995

Recall: 0.3495003065603923

F1: 0.1461094839706049

best result: 0.14943950744647347

singPITF(100轮迭代)：

Precisions: 0.13231707317073252

Recall: 0.5123911149825786

F1: 0.21032179798237877

best result: 0.21468405247488798

movielens core 3 batchsize, 50:

Precisions: 0.13475609756097642

Recall: 0.5280015969802555

F1: 0.21471326640507574

best result: 0.22465817855853398

LastFM core 1:

Sotfattention model gamma=0.5, m=8

TimeAttention gamma=0.6, m=5

**LastFM core 3:**

Softattention model gamma=0.5, m=8

Precisions: 0.2499608457321823

Recall: 0.5221692573427889

F1: 0.3380825813187491

best result: 0.3439032026968862

TimeAttention gamma=0.6, m=5:

Precisions: 0.2499608457321823

Recall: 0.5221692573427889

F1: 0.3380825813187491

best result: 0.3439032026968862

SingPITF**:**

Precisions: 0.2543461237274839

Recall: 0.5323748965656558

F1: 0.34423254957860566

best result: 0.3501683774662842

Delicious core1:

Softattention model gamma=0.5, m=8

TimeAttention gamma=0.6, m=5:

Delicious core 3:

**Softattention model gamma=0.5, m=8**

Precisions: 0.2045267489711915

Recall: 0.30054592307365957

F1: 0.24340925163871396

best result: 0.24614847145505606

batch\_size: 10:

TimeAttention gamma=0.6, m=5:

Precisions: 0.2155006858710541

Recall: 0.31280403707098875

F1: 0.25519167860790753

best result: 0.25606539624730723

SingPITF:

Precisions: 0.22126200274348212

Recall: 0.32397148318266245

F1: 0.2629426880449603

best result: 0.2684144941293935

**根据小学长的建议，我们调整attention策略  
1. 改为tag-attention， 并且将当前tag与历史行为进行attention之后，使用单层网络映射到隐向量维度（batch\_size 调整为32）**

Tag query:

Precisions: 0.005487804878048782

Recall: 0.013236788617886177

F1: 0.007758877453089322

best result: 0.014997647243353462 (代码出错)

User query:

Precisions: 0.08567073170731741

Recall: 0.3606453252032521

F1: 0.13845232954805828

best result: 0.16559334927868646

**使用预训练向量（统一m=8）：**

User query, attention映射，batch\_size 100, learnrate = 0.001

Precisions: 0.12987804878048867

Recall: 0.515962543554007

F1: 0.20751934516344295

best result: 0.2227856455151393

User query, attention映射，batch\_size 100, learnrate = 0.01

Precisions: 0.11310975609756163

Recall: 0.45798127177700354

F1: 0.1814146866944855

best result: 0.22085537393215776

User query，attention 映射+relu 激活 batch\_size 100, learnrate =0.001:

Precisions: 0.1329268292682936

Recall: 0.5263792102206737

F1: 0.21225323360185475

best result: 0.22054163696371668

User query, attention 相减、相减 映射，batch\_siuze 100, learnrate=0.001:

Precisions: 0.12957317073170818

Recall: 0.5140570557491291

F1: 0.20697599307792686

best result: 0.2192025997118584

User query, attention 相减、相减 映射+relu激活，batch\_siuze 100, learnrate=0.001:

Precisions: 0.1329268292682936

Recall: 0.5263792102206737

F1: 0.21225323360185475

best result: 0.22054163696371668

User query,RNN attention 映射+relu激活，batch\_size100, learnrate=0.01:

Precisions: 0.08445121951219535

Recall: 0.30790868176538905

F1: 0.13254801822922746

best result: 0.1513872120011506 (为什么这个结果这么差？存疑)

user query, attention 相乘、相减、映射+relu激活，batchsize100, 学习速率0.001，平均loss：

Precisions: 0.12865853658536663

Recall: 0.4921058362369339

F1: 0.20398598729984171

best result: 0.20451102658743098

tag query, attention, 相乘、相减，映射+relu激活，batchsize100,学习速率0.001：

Precisions: 0.13292682926829355

Recall: 0.5242704703832752

F1: 0.2120812466636532

best result: 0.22716343668625677

RNN,user query, attention,相减，相乘，映射+relu激活，batch size100,

学习速率0.001：

Precisions: 0.13018292682926916

Recall: 0.5190113240418118

F1: 0.20815468753963084

best result: 0.2166719467907358

Tag query, attention, 相减，相乘，映射+relu激活，batch size 100,学习速率0.001， m=5:

Precisions: 0.13292682926829355

Recall: 0.5270143728222997

F1: 0.21230482150884084

best result: 0.22669272374732236

Tag query, attention, 对tag\_history map，相减、相乘、映射+relu激活，batch size 100, 学习速率0.001

Precisions: 0.13109756097561062

Recall: 0.5210438443670151

F1: 0.20948701186051363

best result: 0.22358959956766147

TimeAttention, user query, 映射+relu激活，batch size100 从此开始，学习速率默认为0.001

Precisions: 0.13048780487804965

Recall: 0.5205357142857143

F1: 0.20866700117076545

best result: 0.2216147082827993

User query, attention,相减、相乘、映射+relu激活，embeeding层加入dropout:

Precisions: 0.1329268292682936

Recall: 0.5226190476190478

F1: 0.21194578553395169

best result: 0.22082788385062518

user query, attention, 相减、相乘、映射+relu 激活，embedding=0.3:

Precisions: 0.13262195121951306

Recall: 0.5232796167247388

F1: 0.21161212960946083

best result: 0.21824665747316419

TimeAttention, user query， 映射+relu激活， batch size 100, m=8

Precisions: 0.13048780487804962

Recall: 0.5177409988385597

F1: 0.20844148253352743

best result: 0.21828186757548487

TimeAttention, user query , 相减、相乘、映射+relu激活，：

Precisions: 0.13201219512195211

Recall: 0.52409262485482

F1: 0.21090111137046652

best result: 0.2192555532863689

Tag query, attention, 相减、相乘、映射+relu激活，RNN:

Precisions: 0.1280487804878057

Recall: 0.5116434378629501

F1: 0.20483387599070596

best result: 0.21854006568322595

Tag query, attention, 相减、相乘、映射+relu激活，修来负例attention BUG:

Precisions: 0.12896341463414718

Recall: 0.514057055749129

F1: 0.2061973336141805

best result: 0.21991769427697883

**截止于2018年10月12日，并未在之前各种实验与trick中取得超越TAPITF的结果，因此决定再使用TAPITF的基础之上进行优化**

**临时结果（NeuralTAPITF）:**

Precisions: 0.14634146341463522

Recall: 0.5786403890824623

F1: 0.23360331306908255

MRR: 0.3746104142911438

NDCG: 0.4774373028967218

best result: 0.24388647425389007

TAPITF:

[16/100] [000/2993] mean\_loss : 0.05

[16/100] [1000/2993] mean\_loss : 0.10

[16/100] [2000/2993] mean\_loss : 0.12

Precisions: 0.1570121951219524

Recall: 0.6084168118466899

F1: 0.24960867254162544

MRR: 0.3986697073356134

NDCG: 0.5023135805740796

best result: 0.24960867254162544

Precisions: 0.1551829268292694

Recall: 0.6029544134727062

F1: 0.24683715114216806

MRR: 0.4001710376282779

NDCG: 0.5025385918864543

best result: 0.24960867254162544

tag+ MLP 与 User query结果：

Precisions: 0.1466463414634157

Recall: 0.5820702671312427

F1: 0.23427070041410197

MRR: 0.3709045990117823

NDCG: 0.4748870338919867

best result: 0.24170379761011676

Precisions: 0.15152439024390355

Recall: 0.5957389663182348

F1: 0.2415988495172455

MRR: 0.38458760927403984

NDCG: 0.48660664579475343

best result: 0.24306442286736957

**2018年10月25日，融合神经协同过滤的思想，再试试：**

**user-tag mlp, item-tag mlp user-item-tag mlp 预测层sigmod函数：**

**学习率0.01,10个负例**

Precisions: 0.1371951219512204

Recall: 0.5368466898954704

F1: 0.21854058841699928

MRR: 0.34990497909540086

NDCG: 0.4401761936806795

best result: 0.22314848555858594

学习率0.001,10个负例：

Precisions: 0.11981707317073237

Recall: 0.47118902439024374

F1: 0.19105213988688846

MRR: 0.30993918662105663

NDCG: 0.39259529555658024

best result: 0.1939594127500743

**user-tag mlp, item-tag mlp user-item-tag mlp 预测层relu函数：**

**user-tag mlp, item-tag mlp user-item-tag mlp 预测层不用激活函数：**