Running NeuMF.py with lr=0.0001, num\_factors=16, layers=[16,16]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=200, layers='[16,16]', learner='adam', lr=0.0001, mf\_pretrain='', mlp\_pretrain='', num\_factors=16, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

Load data done [3.3 s]. #user=999, #item=6954, #train=106055, #test=17306

End. Best Iteration 93: HR = 0.4631, NDCG = 0.3116.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_16\_[16,16]\_1732011227.h5

Running NeuMF.py with lr=0.0001, num\_factors=64, layers=[64,64]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=200, layers='[64,64]', learner='adam', lr=0.0001, mf\_pretrain='', mlp\_pretrain='', num\_factors=64, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

Load data done [3.3 s]. #user=999, #item=6954, #train=106055, #test=17306

End. Best Iteration 49: HR = 0.4646, NDCG = 0.3136.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_64\_[64,64]\_1732014314.h5

Running NeuMF.py with lr=0.0001, num\_factors=8, layers=[8,8]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=200, layers='[8,8]', learner='adam', lr=0.0001, mf\_pretrain='', mlp\_pretrain='', num\_factors=8, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 115: HR = 0.4506, NDCG = 0.3062.

Running NeuMF.py with lr=0.0001, num\_factors=32, layers=[32,32]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[32,32]', learner='adam', lr=0.0001, mf\_pretrain='', mlp\_pretrain='', num\_factors=32, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 63: HR = 0.4655, NDCG = 0.3110.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_32\_[32,32]\_1732073147.h5

Running NeuMF.py with lr=0.0005, num\_factors=64, layers=[64,64]

Using Theano backend.

End. Best Iteration 17: HR = 0.4654, NDCG = 0.3144.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_16\_[16,16]\_1732024672.h5

Running NeuMF.py with lr=0.0005, num\_factors=64, layers=[64,64]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[64,64]', learner='adam', lr=0.0005, mf\_pretrain='', mlp\_pretrain='', num\_factors=64, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 7: HR = 0.4700, NDCG = 0.3216.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_64\_[64,64]\_1732026206.h5

Running NeuMF.py with lr=0.0005, num\_factors=8, layers=[8,8]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[8,8]', learner='adam', lr=0.0005, mf\_pretrain='', mlp\_pretrain='', num\_factors=8, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 30: HR = 0.4498, NDCG = 0.2998.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_8\_[8,8]\_1732028795.h5

Running NeuMF.py with lr=0.0005, num\_factors=32, layers=[32,32]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[32,32]', learner='adam', lr=0.0005, mf\_pretrain='', mlp\_pretrain='', num\_factors=32, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 8: HR = 0.4676, NDCG = 0.3195.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_32\_[32,32]\_1732030228.h5

Running NeuMF.py with lr=0.001, num\_factors=16, layers=[16,16]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[16,16]', learner='adam', lr=0.001, mf\_pretrain='', mlp\_pretrain='', num\_factors=16, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 6: HR = 0.4615, NDCG = 0.3132.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_16\_[16,16]\_1732032233.h5

Running NeuMF.py with lr=0.001, num\_factors=64, layers=[64,64]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[64,64]', learner='adam', lr=0.001, mf\_pretrain='', mlp\_pretrain='', num\_factors=64, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 4: HR = 0.4698, NDCG = 0.3172.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_64\_[64,64]\_1732033747.h5

Running NeuMF.py with lr=0.001, num\_factors=8, layers=[8,8]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[8,8]', learner='adam', lr=0.001, mf\_pretrain='', mlp\_pretrain='', num\_factors=8, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 10: HR = 0.4549, NDCG = 0.3086.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_8\_[8,8]\_1732036310.h5

Running NeuMF.py with lr=0.001, num\_factors=32, layers=[32,32]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[32,32]', learner='adam', lr=0.001, mf\_pretrain='', mlp\_pretrain='', num\_factors=32, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 5: HR = 0.4661, NDCG = 0.3174.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_32\_[32,32]\_1732037757.h5

Running NeuMF.py with lr=0.005, num\_factors=16, layers=[16,16]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[16,16]', learner='adam', lr=0.005, mf\_pretrain='', mlp\_pretrain='', num\_factors=16, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

Load data done [3.0 s]. #user=999, #item=6954, #train=106055, #test=17306

Init: HR = 0.1018, NDCG = 0.0470

End. Best Iteration 1: HR = 0.4445, NDCG = 0.2996.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_16\_[16,16]\_1732039786.h5

Running NeuMF.py with lr=0.005, num\_factors=64, layers=[64,64]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[64,64]', learner='adam', lr=0.005, mf\_pretrain='', mlp\_pretrain='', num\_factors=64, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 1: HR = 0.4311, NDCG = 0.2763.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_64\_[64,64]\_1732041307.h5

Running NeuMF.py with lr=0.005, num\_factors=8, layers=[8,8]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[8,8]', learner='adam', lr=0.005, mf\_pretrain='', mlp\_pretrain='', num\_factors=8, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 1: HR = 0.4337, NDCG = 0.2968.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_8\_[8,8]\_1732043864.h5

Running NeuMF.py with lr=0.005, num\_factors=32, layers=[32,32]

Using Theano backend.

NeuMF arguments: Namespace(batch\_size=1024, dataset='amazon\_small', epochs=100, layers='[32,32]', learner='adam', lr=0.005, mf\_pretrain='', mlp\_pretrain='', num\_factors=32, num\_neg=4, out=1, path='Data/', reg\_layers='[0,0]', reg\_mf=0.0, verbose=1)

End. Best Iteration 1: HR = 0.4389, NDCG = 0.2869.

The best NeuMF model is saved to Pretrain/amazon\_small\_NeuMF\_32\_[32,32]\_1732045327.h5