File - distmult

```
ssh://sangsq@10.50.221.195:22/home/sangsq/anaconda3/bin/python -u /home/sangsq/flower/PSEP/model/distmult.py
loading data
remove  0  isolated drugs:  []
remove finished
963  polypharmacy side effects
data has been loaded
cuda
model training ...
  0   loss:1.4452   auprc:0.5035   auroc:0.5000   ap@50:0.5041
  0   loss:1.4452   auprc:0.5048   auroc:0.5000   ap@50:0.5085    time:41.3

File: /home/sangsq/flower/PSEP/model/distmult.py
Function: train at line 94

Line # Max usage   Peak usage diff max diff peak  Line Contents
================================================================
    94                                           @profile
    95                                           def train():
    96   260.34M      276.00M    0.00B  -14.00M      model.train()
    97
    98   260.34M      276.00M    0.00B    0.00B      optimizer.zero_grad()
    99   260.38M      276.00M   40.50K    0.00B      z = model.encoder(data.d_feat, data.train_idx, data.train_et, data.train_range, data.x_norm)
   100
   101   260.38M      276.00M    0.00B    0.00B      pos_index = data.train_idx
   102   386.38M      530.00M  126.00M  254.00M      neg_index = negative_sampling(data.train_idx, n_drug).to(device)
   103
   104     2.37G        2.88G    1.99G    2.37G      pos_score = model.decoder(z, pos_index, data.train_et)
   105     4.37G        5.31G    1.99G    2.43G      neg_score = model.decoder(z, neg_index, data.train_et)
   106
   107                                               # pos_loss = F.binary_cross_entropy(pos_score, torch.ones(pos_score.shape[0]).cuda())
   108                                               # neg_loss = F.binary_cross_entropy(neg_score, torch.ones(neg_score.shape[0]).cuda())
   109     4.40G        4.82G   31.42M -504.00M      pos_loss = -torch.log(pos_score + EPS).mean()
   110     4.43G        4.82G   31.42M    0.00B      neg_loss = -torch.log(1 - neg_score + EPS).mean()
   111     4.43G        4.82G  512.00B    0.00B      loss = pos_loss + neg_loss
   112                                               # loss = pos_loss
   113
   114   449.33M        9.25G   -3.99G    4.42G      loss.backward()
   115   449.52M      906.00M  202.00K   -8.36G      optimizer.step()
   116
   117   449.52M      906.00M    0.00B    0.00B      record = np.zeros((3, n_et_dd))  # auprc, auroc, ap
   118   449.92M      906.00M  401.50K    0.00B      for i in range(data.train_range.shape[0]):
   119   449.92M      906.00M    0.00B    0.00B          [start, end] = data.train_range[i]
   120   449.92M      906.00M    0.00B    0.00B          p_s = pos_score[start: end]
   121   449.92M      906.00M    0.00B    0.00B          n_s = neg_score[start: end]
   122
   123   449.92M      906.00M    0.00B    0.00B          pos_target = torch.ones(p_s.shape[0])
   124   449.92M      906.00M    0.00B    0.00B          neg_target = torch.zeros(n_s.shape[0])
   125
```

```
49    126   449.92M      906.00M    0.00B    0.00B          score = torch.cat([p_s, n_s
   ])
50    127   449.92M      906.00M    0.00B    0.00B          target = torch.cat([
   pos_target, neg_target])
51    128
52    129   449.92M      906.00M    0.00B    0.00B          record[0, i], record[1, i
   ], record[2, i] = auprc_auroc_ap(target, score)
53    130
54    131   449.53M      906.00M -394.00K    0.00B      train_record[epoch] = record
55    132   449.53M      906.00M    0.00B    0.00B      [auprc, auroc, ap] = record.sum
   (axis=1) / n_et_dd
56    133   449.53M      906.00M    0.00B    0.00B      train_out[epoch] = [auprc,
   auroc, ap]
57    134
58    135   449.53M      906.00M    0.00B    0.00B      print('{:3d}   loss:{:0.4f}
   auprc:{:0.4f}   auroc:{:0.4f}   ap@50:{:0.4f}'
59    136   449.53M      906.00M    0.00B    0.00B            .format(epoch, loss.
   tolist(), auprc, auroc, ap))
60    137
61    138   449.53M      906.00M    0.00B    0.00B      return z, loss
62
63
64 Process finished with exit code 0
65
```