

به نام خدا



دانشگاه تهران



دانشکده مهندسی برق و کامپیوتر

**درس شبکه‌های عصبی و یادگیری عمیق**

**تمرین اول**

نام و نام خانوادگی	فاطمه نائینیان – محمد عبائینی
شماره دانشجویی	810198432 - 810198479
تاریخ ارسال گزارش	1401.08.19

## فهرست

3 ..... Restricted Boltzmann Machine – پاسخ ۳

## پاسخ ۳ – Restricted Boltzmann Machine

(A)

در این بخش ابتدا با کمک کتابخانه pandas دو csv امتیاز ها و فیلم ها را لود میکنیم.  
حال با دستور head() میتوانیم 5 مورد اول دیتا ست و با دستور tail() 5 مورد آخر دیتا ست را مشاهده میکنیم.

```
movies.head()
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
1	2	Jumanji (1995)	Adventure Children Fantasy
2	3	Grumpier Old Men (1995)	Comedy Romance
3	4	Waiting to Exhale (1995)	Comedy Drama Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
movies.tail()
```

	movieId	title	genres
9737	193581	Black Butler: Book of the Atlantic (2017)	Action Animation Comedy Fantasy
9738	193583	No Game No Life: Zero (2017)	Animation Comedy Fantasy
9739	193585	Flint (2017)	Drama
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action Animation
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy

```
ratings.head()
```

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

```
ratings.tail()
```

	userId	movieId	rating	timestamp
100831	610	166534	4.0	1493848402
100832	610	168248	5.0	1493850091
100833	610	168250	5.0	1494273047
100834	610	168252	5.0	1493846352
100835	610	170875	3.0	1493846415

با دستور shape ابعاد دیتا ست ها را میتوانیم ببینیم.

```
print(movies.shape)  
print(ratings.shape)
```

```
(9742, 3)  
(100836, 4)
```

حال ستونی به نام index در دیتاست فیلم ها درست میکنیم و مقدار ان را برابر index ست میکنیم.

```
movies['List Index'] = movies.index
```

(B)

حال دو دیتاست را بر اساس ستون movieId ادغام میکنیم. اینکار با کمک تابع merge صورت میگیرد.

```
df = pd.merge(movies,ratings,on = "movieId")
df
```

	movieId	title	genres	List Index	userId	rating	timestamp
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0	1	4.0	964982703
1	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0	5	4.0	847434962
2	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0	7	4.5	1106635946
3	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0	15	2.5	1510577970
4	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0	17	4.5	1305696483
...	...	...	...	...	...	...	...

(C)

حال ستون هایی که اضافه هستند را حذف میکنیم.

ستون هایی که از نوع categorical هستند و مقادیر آنها عموماً منحصر به فرد است، اطلاعات زیادی در اختیار ما قرار نمیدهند. در نتیجه آنها را حذف میکنیم. برای نمونه، اسم فیلم هیچ ویژگی ای در اختیار ما نمیگذارد، ژانر نیز در امتیاز دهی تاثیری ندارد و ستون timestamp اشاره به زمان رای دهی دارد. پس اطلاعاتی در بر نداشته و حذف میشوند.

```
del df["title"]
del df["genres"]
del df["timestamp"]
```

```
print(df.head())
```

	movieId	List Index	userId	rating
0	1	0	1	4.0
1	1	0	5	4.0
2	1	0	7	4.5
3	1	0	15	2.5
4	1	0	17	4.5

(D)

حال می‌خواهیم از متد groupby استفاده کنیم و داده‌ها را بر اساس userID گروه‌بندی کنیم. خروجی این تابع شامل 610 ماتریس (به تعداد کاربر‌ها) است که در هر ماتریس فیلم‌هایی که آن کاربر به آنها رای داده را مشاهده می‌کنیم. در حالت کلی همه رای‌ها را در یک دیتا ست به شکل زیر مشاهده می‌کنیم.

```
Grouped_df = df.groupby('userId')
Grouped_df.head()
```

	movieId	List Index	userId	rating
0	1	0	1	4.0
1	1	0	5	4.0
2	1	0	7	4.5
3	1	0	15	2.5
4	1	0	17	4.5
...	...	...	...	...
68658	5816	4076	556	4.5
69478	5989	4159	550	4.0
72615	6874	4615	2	4.0
75076	7444	4939	506	3.0
80944	37741	6010	506	4.0

3050 rows × 4 columns

مثلا برای کاربر شماره 1 داریم:

1				
	movieId	List Index	userId	rating
0	1	0	1	4.0
325	3	2	1	4.0
433	6	5	1	4.0
2107	47	43	1	5.0
2379	50	46	1	5.0
...	...	...	...	...
56816	3744	2802	1	4.0
57276	3793	2836	1	5.0
57457	3809	2847	1	4.0
59170	4006	2991	1	4.0
65533	5060	3673	1	5.0

[232 rows x 4 columns]

(E)

در این بخش میتوانیم با کمک تعریف یک ماتریس که به تعداد کاربر ها سطر دارد و به تعداد فیلم ها ستون دارد ، train\_x را تولید کنیم. ابتدا همه مقادیر این ماتریس را صفر قرار میدهیم سپس از خروجی groupby استفاده میکنیم و امتیاز هر فیلم را در جایگاهش در ماتریس قرار میدهیم. بنابر این امتیاز صفر را به عنوان عدم رای دهی قرار دادیم. بدین ترتیب یک لیست از امتیازات خواهیم داشت.

```
1 Users_num = 610
2 train_X = []
3 for userId, curUser in Grouped_df:
4
5     temp = [0]*len(movies)
6     for num, movie in curUser.iterrows():
7         temp[int(movie['List Index'])] = movie['rating']/5.0
8     train_X.append(temp)
9 print(train_X)
10 train_X = torch.FloatTensor(train_X)
```

(F)

تعداد نورون های visible layer به تعداد فیلم ها یعنی 9742 است و تعداد نورون های hidden layer برابر 20 است.

بدین ترتیب یک class RBM تعریف میکنیم که چهار تابع اصلی دارد. تابع init که مقادیر اولیه کلاس را میگیرد. تابع sample\_h که به ازای یک کاربر خاص مقادیر hidden layer را محاسبه میکند یعنی ضرب  $xw+b$  را انجام داده و از activation function عبور میدهد. تابع sample\_h که به ازای یک کاربر خاص از hidden layer به visible layer برمیگردد و باز هم مقادیر را حساب میکند و در نهایت از یک تابع sigmoid عبور داده میشود. تابع train که وظیفه آموزش هایپر پارامتر ها را دارد و به روشی که در درس آموزش داده شد هایپر پارامتر ها را اپدیت میکند.

```

class RBM():
    def __init__(self, nv, nh,alpha):
        self.W = torch.randn(nh, nv)
        self.a = torch.randn(1, nh)
        self.b = torch.randn(1, nv)
        self.alpha = alpha
    def sample_h(self, x):
        wx = torch.mm(x, self.W.t())
        activation = wx + self.a.expand_as(wx)
        p_h_given_v = torch.sigmoid(activation)
        return p_h_given_v, torch.bernoulli(p_h_given_v)
    def sample_v(self, y):
        wy = torch.mm(y, self.W)
        activation = wy + self.b.expand_as(wy)
        p_v_given_h = torch.sigmoid(activation)
        return p_v_given_h, torch.bernoulli(p_v_given_h)
    def train(self, v0, vk, ph0, phk):
        self.W += (torch.mm(v0.t(), ph0) - torch.mm(vk.t(), phk)).t()
        self.b += self.alpha*torch.sum((v0 - vk), 0)
        self.a += self.alpha*torch.sum((ph0 - phk), 0)

```

حال با کمک یک for به تعداد ایپاک های خواسته شده با حالت batch size = 100 داده ها را آموزش می‌دهیم و خطا را به روش mean absolute error حساب می‌کنیم.

```

nb_epoch = 20
nb_users = len(train_X)
loss = []
for epoch in range(1, nb_epoch + 1):
    train_loss = 0
    s = 0.
    for id_user in range(0, nb_users - batch_size, batch_size):
        vk = train_X[id_user:id_user+batch_size]
        v0 = train_X[id_user:id_user+batch_size]
        ph0,_ = rbm.sample_h(v0)
        for k in range(10):
            _,hk= rbm.sample_h(vk)
            _,vk = rbm.sample_v(hk)
            vk[v0<0] = v0[v0<0]
            phk,_ = rbm.sample_h(vk)
            rbm.train(v0, vk, ph0, phk)
            train_loss += torch.mean(torch.abs(v0[v0>=0] - vk[v0>=0]))
        s += 1.
    print('epoch: '+str(epoch)+' loss: '+str(train_loss/s))
    loss.append(train_loss.float()/s)
plt.plot(loss)
plt.title('loss')
plt.xlabel('epoch')

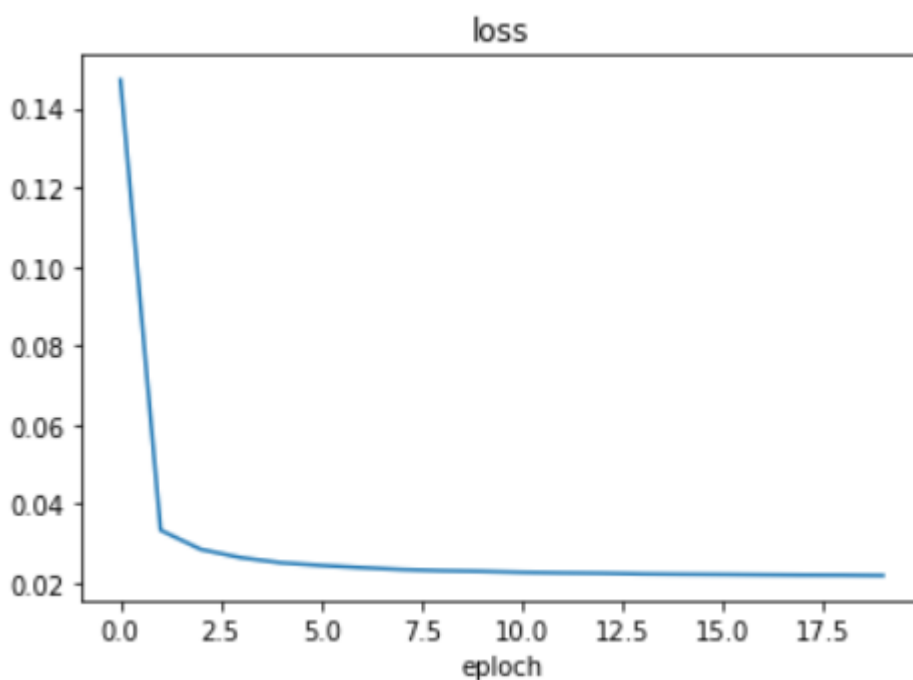
```



(G)

مدل را به مدت 20 اپاک ترین میکنیم و نمودار loss به شکل زیر به دست می آید:

```
epoch: 1 loss: tensor(0.1472)
epoch: 2 loss: tensor(0.0332)
epoch: 3 loss: tensor(0.0284)
epoch: 4 loss: tensor(0.0263)
epoch: 5 loss: tensor(0.0250)
epoch: 6 loss: tensor(0.0243)
epoch: 7 loss: tensor(0.0238)
epoch: 8 loss: tensor(0.0233)
epoch: 9 loss: tensor(0.0230)
epoch: 10 loss: tensor(0.0228)
epoch: 11 loss: tensor(0.0226)
epoch: 12 loss: tensor(0.0224)
epoch: 13 loss: tensor(0.0224)
epoch: 14 loss: tensor(0.0222)
epoch: 15 loss: tensor(0.0221)
epoch: 16 loss: tensor(0.0220)
epoch: 17 loss: tensor(0.0220)
epoch: 18 loss: tensor(0.0219)
epoch: 19 loss: tensor(0.0219)
epoch: 20 loss: tensor(0.0218)
```



میبینیم در نهایت در حدود 98 درصد دقت داریم.

(H)

در نهایت مدل ممکن است تعداد زیادی فیلم پیشنهاد دهد بنابراین به ازای یک کاربر خاص میتوانیم جایگشت های بسیار زیادی از فیلم های پیشنهاد شده را مشاهده کنیم.

برای مثال برای کاربر 75 ام داریم:

```

movieId          title          genres List Index
3144      4231  Someone Like You (2001) Comedy|Romance      3144
movieId          title          genres List Index
995       1297  Real Genius (1985) Comedy      995
movieId          title \
1979      2628  Star Wars: Episode I - The Phantom Menace (1999)

genres List Index
1979  Action|Adventure|Sci-Fi      1979
movieId          title          genres List Index
2553      3418  Thelma & Louise (1991) Adventure|Crime|Drama      2553
movieId          title          genres List Index
7655     88405  Friends with Benefits (2011) Comedy|Romance      7655
movieId          title          genres List Index
2555      3421  Animal House (1978) Comedy      2555
movieId          title          genres List Index
594       736  Twister (1996) Action|Adventure|Romance|Thriller      594
movieId          title          genres List Index
4948     7458  Troy (2004) Action|Adventure|Drama|War      4948
movieId          title          genres List Index
464       531  Secret Garden, The (1993) Children|Drama      464
movieId          title          genres List Index
896      1193  One Flew Over the Cuckoo's Nest (1975) Drama      896
movieId          title          genres \
1480      2005  Goonies, The (1985) Action|Adventure|Children|Comedy|Fantasy

List Index
1480      1480
movieId          title          genres \
5263      8641  Anchorman: The Legend of Ron Burgundy (2004) Comedy

List Index
5263      5263
movieId          title          genres List Index
587       724  Craft, The (1996) Drama|Fantasy|Horror|Thriller      587
movieId          title          genres List Index
84        95  Broken Arrow (1996) Action|Adventure|Thriller      84

```

برای کاربر 450 داریم:

```

movieId      title      genres List Index
1476      2001  Lethal Weapon 2 (1989)  Action|Comedy|Crime|Drama      1476
movieId      title      genres List Index
3016      4034  Traffic (2000)  Crime|Drama|Thriller      3016
movieId      title      genres \
990      1291  Indiana Jones and the Last Crusade (1989)  Action|Adventure

List Index
990      990
movieId      title      genres List Index
2302      3052  Dogma (1999)  Adventure|Comedy|Fantasy      2302
movieId      title      genres List Index
7965      96417  Premium Rush (2012)  Action|Thriller      7965
movieId      title      genres List Index
9151      147384  Doctor Who: The Runaway Bride (2007)  Sci-Fi      9151
movieId      title      genres List Index
4764      7090  Hero (Ying xiong) (2002)  Action|Adventure|Drama      4764
movieId      title      genres List Index
6710      58559  Dark Knight, The (2008)  Action|Crime|Drama|IMAX      6710
movieId      title      genres List Index
5627      27251  10th Kingdom, The (2000)  Adventure|Comedy|Fantasy      5627
movieId      title      genres \
3010      4027  O Brother, Where Art Thou? (2000)  Adventure|Comedy|Crime

List Index
3010      3010
movieId      title \
1972      2617  Mummy, The (1999)

... ..
genres List Index
1972  Action|Adventure|Comedy|Fantasy|Horror|Thriller      1972
movieId      title      genres List Index
884      1179  Grifters, The (1990)  Crime|Drama|Film-Noir      884
movieId      title      genres List Index
2372      3147  Green Mile, The (1999)  Crime|Drama      2372
movieId      title \
8802      130482  Too Late for Tears (1949)

genres List Index
8802  Crime|Drama|Film-Noir|Mystery|Thriller      8802

```