



به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر



درس شبکه‌های عصبی و یادگیری عمیق

تمرین اول

نام و نام خانوادگی	مهرسا ندادفی
شماره دانشجویی	810100490
تاریخ ارسال گزارش	۱۴۰۱.۰۷.۰۱

فهرست

- 5.....**Restricted Boltzmann Machine – ۳**
5.....۳-۱. سیستم توصیه گر

شکل‌ها

- 5..... شکل 1- 5 مورد اول و آخر دیتاست movies
- 6..... شکل 2- 5 مورد اول و آخر دیتاست ratings
- 7..... شکل 3- افزودن movies به دیتاست List Index
- 7..... شکل 4- تلفیق دو دیتاست ratings
- 8..... شکل 5- گروه بندی دیتاست ratings
- 9..... شکل 6- مدل RBM
- 9..... شکل 7- فرمولهای استفاده شده در RBM
- 11..... شکل 8- نمودار Loss در 20 ایپاک
- 12..... شکل 9- 15 فیلم برتر امتیاز دهی شده توسط مدل از کاربر

پاسخ ۳ - Restricted Boltzmann Machine

۱-۳. سیستم توصیه گر

(A) در ابتدا دو دیتا ست را خوانده و ۵ مورد اول و آخر به شکل زیر نمایش داده شد:

`▶ 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

شكل ۱-۵ مورد اول و آخر دیتابست `movies` است

The screenshot shows two code cells in a Jupyter Notebook. The first cell contains the command `ratings.head()` and displays the first five rows of the ratings dataset. The second cell contains the command `ratings.tail()` and displays the last five rows of the same dataset.

	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

	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

شكل 2-5 مورد اول و آخر دیتاست ratings

ابعاد هر کدام با shape به صورت زیر است:

shape of movies dataset: (9742, 3)

shape of ratings dataset: (100836, 4)

سپس با استفاده از کد زیر ستون List Index را به دو دیتا اضافه میکنیم:

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

نتیجه این دیتا بصورت زیر خواهد بود:

movieId		title	genres	List Index
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	0
1	2	Jumanji (1995)	Adventure Children Fantasy	1
2	3	Grumpier Old Men (1995)	Comedy Romance	2
3	4	Waiting to Exhale (1995)	Comedy Drama Romance	3
4	5	Father of the Bride Part II (1995)	Comedy	4
...
9737	193581	Black Butler: Book of the Atlantic (2017)	Action Animation Comedy Fantasy	9737
9738	193583	No Game No Life: Zero (2017)	Animation Comedy Fantasy	9738
9739	193585	Flint (2017)	Drama	9739
9740	193587	Bungo Stray Dogs: Dead Apple (2018)	Action Animation	9740
9741	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy	9741

شکل 3- افزودن List Index به دیتاست movies

(B) در این بخش دو دیتا را با استفاده از merge و ستون movieId با استفاده از دستور زیر تلفیق کردیم:

`movies.merge(ratings, on='movieId')`

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
...
100831	193581	Black Butler: Book of the Atlantic (2017)	Action Animation Comedy Fantasy	9737	184	4.0	1537109082
100832	193583	No Game No Life: Zero (2017)	Animation Comedy Fantasy	9738	184	3.5	1537109545
100833	193585	Flint (2017)	Drama	9739	184	3.5	1537109805
100834	193587	Bungo Stray Dogs: Dead Apple (2018)	Action Animation	9740	184	3.5	1537110021
100835	193609	Andrew Dice Clay: Dice Rules (1991)	Comedy	9741	331	4.0	1537157606

شکل 4- تلفیق دو دیتاست

(C) ویژگی های title , genres به علت کلاکتری بودن نمی توانند نقش خوبی ایفا کنند پس حذف میشوند، ویژگی timestamp نیز حذف میشود چراکه مشخصه‌ی مناسبی نیست.

`del df['timestamp'],df['title'],df['genres']`

(D) با استفاده از کد groupby داده ها را با گروه بندی کردیم:

	movieId	List	Index	rating
userId				
1	1	0	4.0	
2	318	277	3.0	
3	31	30	0.5	
4	21	20	3.0	
5	1	0	4.0	

شکل 5- گروه بندی دیتاست ratings

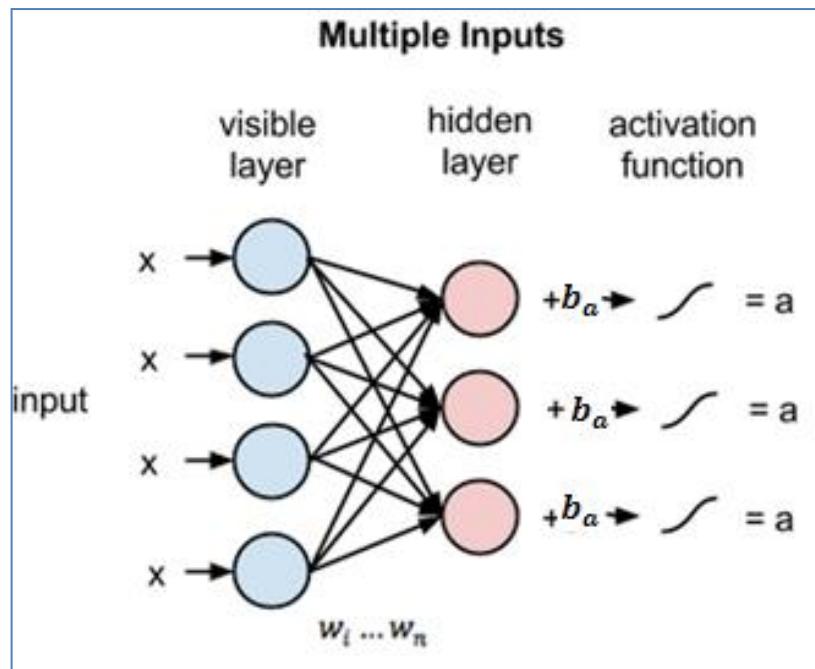
E) در این بخش داده های ستون rating را با استفاده از فرمول زیر نرمالیزه کرده در X_train ذخیره کردیم:

$$\frac{rating[]}{max(rating[])}$$

اینکار را برای هر کاربر و سپس برای تمام فیلمهایی که توسط کاربر امتیازدهی شده است انجام دادیم و فیلمهایی که توسط کاربر امتیاز دهی نشده است صفر در نظر گرفته میشود.

F,G) این شبکه بازنمایی کوچکی از شبکه های Boltzmann Machine که شبکه های آماری هستند می باشد فرم محدود شده آن شامل دو لایه visible , hidden می باشد میتوان برای کاهش ابعاد از آن استفاده کرد چرا که در ورودی که لایه visible است میتواند تعدادی ویژگی را بگیرد و در خروجی که لایه ی hidden است کاهش بعد دهد میتوان از معماری عکس برای رسیدن مجدد به ورودی مذکور استفاده کرد.

در این سوال هدف حدس امتیاز هایی است که توسط کاربران داده نشده است اینکار با استفاده از امتیازدهی دیگر کاربران و یافتن ویژگی های مناسب از آنها توسط ندل انجام میپذیرد.



شکل 6-مدل RBM

1. Learning by Gradient Search

$$J(W, b_a, b_r) = 0.5 \sum_{i=1}^m \|x^i - \sigma(W^T \sigma(Wx^i + b_a) + b_r)\|^2$$

$$W^+ = W^- - \alpha \nabla J_W(W, b_a, b_r)$$

$$b_a^+ = b_a^- - \alpha \nabla J_{b_a}(W, b_a, b_r) \quad b_r^+ = b_r^- - \alpha \nabla J_{b_r}(W, b_a, b_r)$$

شکل 7- فرمولهای استفاده شده در RBM

با استفاده از فرمول های بالا مدل RBM ساخته شد و برای توابع فعال ساز از sigmoid و relu استفاده شد:

```
v = 9742 #visible
h = 20 #hidden

tf.compat.v1.disable_eager_execution()

br = tf.compat.v1.placeholder(tf.float32, [v])
W = tf.compat.v1.placeholder(tf.float32, [v, h])
ba = tf.compat.v1.placeholder(tf.float32, [h])

x = tf.compat.v1.placeholder(tf.float32, [None, v])
```

```

r = tf.nn.relu(tf.sign(tf.nn.sigmoid(tf.matmul(x, W) + ba) - tf.random.uniform(tf.shape(tf.nn.sigmoid(tf.matmul(x, W) + ba))))) #input

a1 = tf.nn.relu(tf.sign(tf.nn.sigmoid(tf.matmul(r, tf.transpose(W)) + br) - tf.random.uniform(tf.shape(tf.nn.sigmoid(tf.matmul(r, tf.transpose(W)) + br))))) #reconstruction

r1 = tf.nn.sigmoid(tf.matmul(a1, W) + ba)

alpha = 0.1#learning rate

g = (tf.matmul(tf.transpose(a1), r1) - tf.matmul(tf.transpose(x), r)) / tf.cast(tf.shape(x)[0], dtype=tf.float32) #gradients + maximize contrastive divergence

update_w = W - alpha*g #update

update_br = br - alpha*tf.reduce_mean(a1 - x, 0)

update_ba = ba - alpha*tf.reduce_mean(r1 - r, 0)

loss = tf.reduce_mean((a1 - x)*(a1 - x)) #MSE

```

با استفاده از 20 ایپاک و batchsize=50 مدل ترین شد:

```

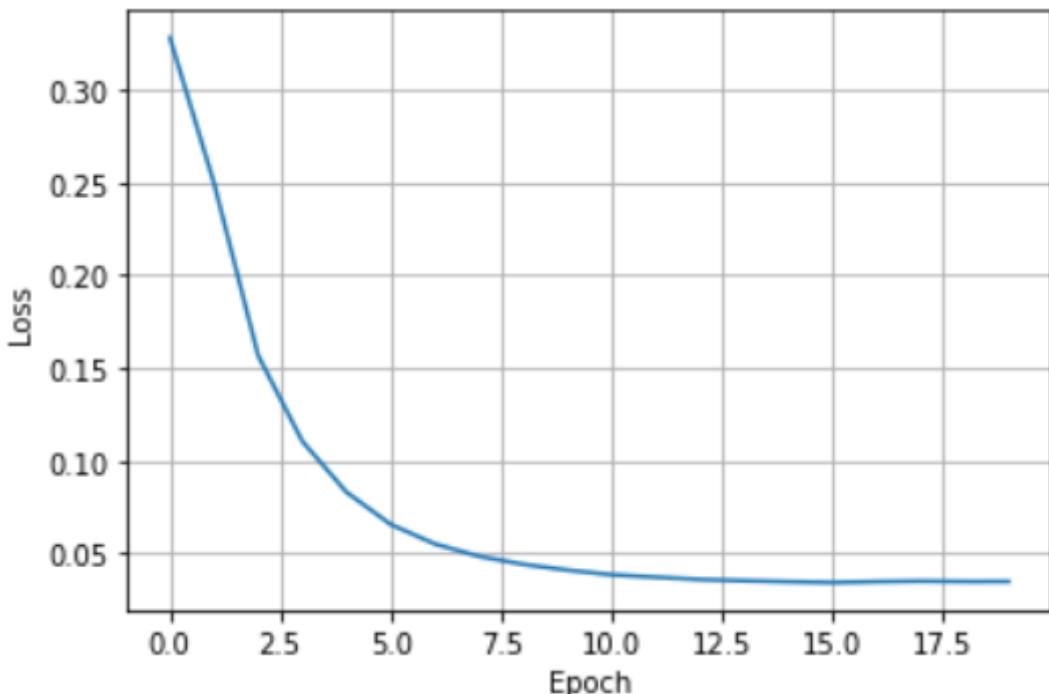
#Current
cw = np.zeros([v, h], np.float32)
cbr = np.zeros([v], np.float32)
cba = np.zeros([h], np.float32)
#Previous
pw = np.zeros([v, h], np.float32)
pbr = np.zeros([v], np.float32)
pba = np.zeros([h], np.float32)
sess = tf.compat.v1.Session()
sess.run(tf.compat.v1.global_variables_initializer())
#Train RBM with 20 epochs and batchsize=50
epochs = 20
batchsize = 50
losses = list()
for i in range(epochs):
    for j, k in zip(range(0, len(X_train), batchsize), range(batchsize, len(X_train), batchsize)):
        batch = X_train[j:k]
        cw = sess.run(update_w, feed_dict={x: batch, W: pw, br: pbr, ba: pba})
        cbr = sess.run(update_br, feed_dict={x: batch, W: pw, br: pbr, ba: pba})
        cnb = sess.run(update_ba, feed_dict={x: batch, W: pw, br: pbr, ba: pba})

```

```

pw = cw
pbr = cbr
pba = cnb
losses.append(sess.run(loss, feed_dict={x: X_train, W: cw, br: cb,
r, ba: cnb}))
```

نمودار loss در هر ایپاک در زیر نشان داده شده است:



شکل 8- نمودار Loss در 20 ایپاک

همانطور که مشخص است مقدار Loss بصورت نمایی کاهش یافته و به مقدار ثابتی رسیده است که نشان از به خوبی ترین شدن مدل است.

(H) در این بخش یوزر 75 از لیست نرمالیزه شده X_{train} انتخاب شده و به عنوان ورودی به شبکه داده شده است سپس خروجی آن که امتیازهای سیستم است در ستون recommendation به دیتابست فیلمها اضافه شده است و سپس sort شده و به شکل زیر نمایش داده شده است:

```

user = [X_train[75]]
r_0 = tf.nn.sigmoid(tf.matmul(x, W) + ba)
r_1 = tf.nn.sigmoid(tf.matmul(r_0, tf.transpose(W)) + br)
feed = sess.run(r_0, feed_dict={x: user, W: pw, ba: pba})
r = sess.run(r_1, feed_dict={r_0: feed, W: pw, br: pbr})
user75 = movies
user75["recommendation"] = r[0]
```

movieId		title	genres	recommendation
277	318	Shawshank Redemption, The (1994)	Crime Drama	0.512960
314	356	Forrest Gump (1994)	Comedy Drama Romance War	0.478854
257	296	Pulp Fiction (1994)	Comedy Crime Drama Thriller	0.394847
1939	2571	Matrix, The (1999)	Action Sci-Fi Thriller	0.376874
97	110	Braveheart (1995)	Action Drama War	0.328509
2226	2959	Fight Club (1999)	Action Crime Drama Thriller	0.324811
224	260	Star Wars: Episode IV - A New Hope (1977)	Action Adventure Sci-Fi	0.317928
510	593	Silence of the Lambs, The (1991)	Crime Horror Thriller	0.304973
418	480	Jurassic Park (1993)	Action Adventure Sci-Fi Thriller	0.298191
461	527	Schindler's List (1993)	Drama War	0.283694
4800	7153	Lord of the Rings: The Return of the King, The...	Action Adventure Drama Fantasy	0.282395
398	457	Fugitive, The (1993)	Thriller	0.266516
506	588	Aladdin (1992)	Adventure Animation Children Comedy Musical	0.253088
2145	2858	American Beauty (1999)	Drama Romance	0.250824
123	150	Apollo 13 (1995)	Adventure Drama IMAX	0.245775

شکل 9-15 فیلم امتیاز دهی شده توسط مدل از کاربر 75

