به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین چهارم

نام و نام خانوادگی	حسام اسداله‌زاده – مسعود طهماسبی
شماره دانشجویی	810198346 - 810198429
تاریخ ارسال گزارش	۱۴۰۱.۰۹.۲۲

فهرست

شکل‌ها

جدول‌ها

پاسخ ۱ – تخمین آلودگی هوا

۱-۱- شرح متدهای مختلف

درون‌یابی خطی (Linear Interpolation) یک روش برای برازش منحنی (Curve Fitting) با استفاده از چند جمله‌ای‌های خطی برای استخراج نقاط جدید است. به طور ساده، می‌توان گفت که «درون یابی خطی» (Linear Interpolation) عبور دادن یک خط راست از بین نقاط داده است.

ضریب همبستگی پیرسون که به نام‌های ضریب همبستگی گشتاوری و یا ضریب همبستگی مرتبه‌ی صفر نیز نامیده می‌شود، توسط کارل پیرسون معرفی شده است. این ضریب به منظور تعیین میزان رابطه، نوع و جهت رابطه‌ی بین دو متغیر فاصله‌ای یا نسبی و یا یک متغیر فاصله‌ای و یک متغیر نسبی به کار برده می‌شود. برای محاسبه‌ی این ضریب با استفاده از داده‌های خام از روابط زیر استفاده می‌شود:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i-1}^{n}(y_i - \bar{y})^2}}$$

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

ضریب همبستگی پیرسون بین 1- و 1 تغییر می کند. $r = 1$ بیانگر رابطه ی مستقیم کامل بین دو متغیر است، رابطه‌ی مستقیم یا مثبت به این معناست که اگر یکی از متغیرها افزایش (یا کاهش) یابد، دیگری نیز افزایش (یا کاهش) می یابد. زمانی که ضریب همبستگی برابر صفر باشد، نشان می‌دهد که بین دو متغیر رابطه‌ی خطی وجود ندارد.

ضریب تعیین (R^2 – squared correlation) میزان ارتباط خطی بین دو متغیر را اندازه گیری می کند. R^2 نسبت تغییرات متغیر وابسته را که می‌توان به متغیر مستقل نسبت داد را اندازه گیری می‌کند. در تعاریف موجود به R^2، ضریب تعیین یا ضریب تشخیص نیز گفته می شود. به بیان ساده می‌توان گفت ضریب تعیین نشان می‌دهد که چند درصد تغییرات متغیرهای وابسته در یک مدل رگرسیونی با متغیر مستقل تبیین می شود. به عبارت دیگر، ضریب تشخیص یا (R^2) نشان می دهد که چه میزان یا مقدار از تغییرات متغیر وابسته مساله تحت تاثیر متغیر مستقل مساله بوده است. همچنین تا چه حدی مابقی

تغییرات متغیر وابسته مساله مربوط به سایر عوامل موجود در مساله است. ضریب تعیین امتیاز» با افزایش، دقت بالای مدل را نشان می‌دهد. برای محاسبه آن به شکل زیر عمل می‌کنیم:

$$R^2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2}$$

این معیار همواره عددی کوچک‌تر از 1 است. اگر مدلی همواره میانگین ویژگی هدف را در خروجی تولید کند، مقدار این معیار برابر 0 خواهد بود. از ضریب تعیین برای مقایسه مدل‌ها و گزارش نتایج استفاده می‌شود. با توجه به اینکه واریانس مجموعه داده‌ی عددی ثابت است، با افزایش میانگین مربعات خطا، ضریب تعیین همواره کاهش می‌یابد.

Missing values –۱–۳–۱

تعداد داده‌های گمشده در هر ستون:

```
[7]  1 dfs['Aotizhongxin'].isna().sum()

No            0
year          0
month         0
day           0
hour          0
PM2.5       925
PM10        718
SO2         935
NO2        1023
CO         1776
O3         1719
TEMP         20
PRES         20
DEWP         20
RAIN         20
wd           81
WSPM         14
station       0
dtype: int64
```

جاگذاری داده‌های گمشده:

```
[10]  1 dfs['Aotizhongxin'].interpolate(method ='linear', inplace=True)

      1 dfs['Aotizhongxin'].isna().sum()

No          0
year        0
month       0
day         0
hour        0
PM2.5       0
PM10        0
SO2         0
NO2         0
CO          0
O3          0
TEMP        0
PRES        0
DEWP        0
RAIN        0
wd          0
WSPM        0
station     0
dtype: int64
```

همچنین این کار را برای ستون PM2.5 سایر سایت‌ها نیز انجام می‌دهیم:

```
[12]  1 for site in sites:
      2     print(site)
      3     print('Before Interpolation', dfs[site]['PM2.5'].isna().sum())
      4     dfs[site]['PM2.5'].interpolate(method='linear', inplace=True)
      5     print('After Interpolation:', dfs[site]['PM2.5'].isna().sum())
      6     print('-----------------------------------------------')
```

```
Gucheng
Before Interpolation 646
After Interpolation: 0
-----------------------------------------------
Tiantan
Before Interpolation 677
After Interpolation: 0
-----------------------------------------------
Dongsi
Before Interpolation 750
After Interpolation: 0
-----------------------------------------------
Guanyuan
Before Interpolation 616
After Interpolation: 0
-----------------------------------------------
Nongzhanguan
Before Interpolation 628
After Interpolation: 0
-----------------------------------------------
```

۱-۳-۲- Encoding Categorical Variable

```
[8]   1 wind_dict = {'N': 0., 'NNE': 22.5, 'NE':45., 'ENE': 67.5, 'E': 90.,
      2              'ESE':112.5, 'SE':135., 'SSE':157.5, 'S':180., 'SSW':202.5,
      3              'SW':225, 'WSW':247.5, 'W':270., 'WNW':292.5, 'NW':315,
      4              'NNW':337.5, 'N':360.}
```

```
[9]   1 for site in sites:
      2     dfs[site]['wd'] = dfs[site]['wd'].map(wind_dict)
      3 dfs['Aotizhongxin']['wd']
```

```
0        337.5
1        360.0
2        337.5
3        315.0
4        360.0
         ...
35059    315.0
35060    292.5
35061    315.0
35062    337.5
35063     22.5
Name: wd, Length: 35064, dtype: float64
```

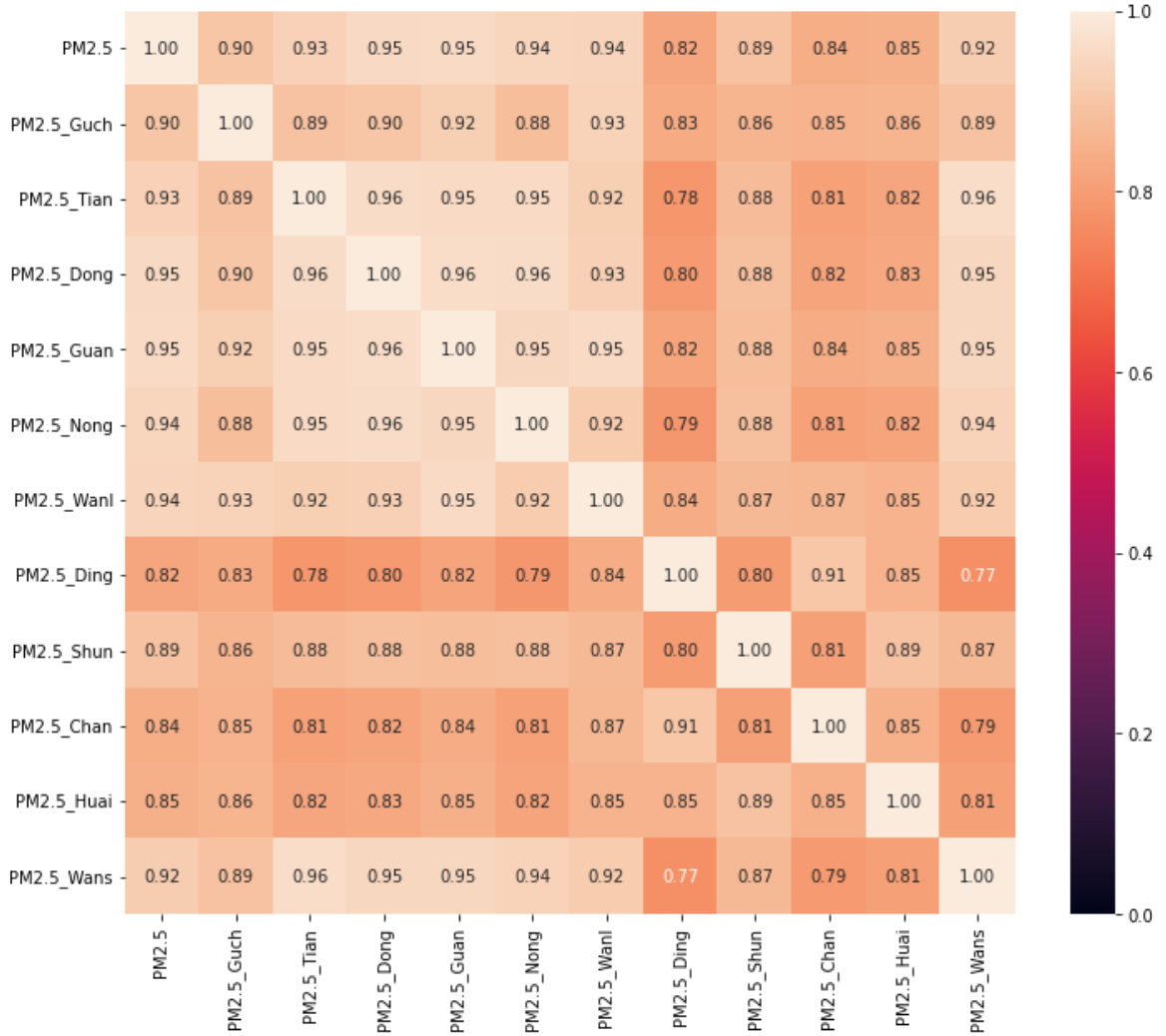
۱-۳-۳- Normalization

از روش Min-Max Normalizartion استفاده می‌کنیم:

```
[13]  1 for site in sites:
      2     cmin = dfs[site]['PM2.5'].min()
      3     cmax = dfs[site]['PM2.5'].max()
      4     dfs[site]['PM2.5'] = (dfs[site]['PM2.5']-cmin)/(cmax-cmin)
      5 dfs['Aotizhongxin']['PM2.5']
```

```
0        0.001117
1        0.005587
2        0.004469
3        0.003352
4        0.000000
           ...
35059    0.010056
35060    0.011173
35061    0.014525
35062    0.020112
35063    0.017877
Name: PM2.5, Length: 35064, dtype: float64
```

Pearson Correlation –۴–۳–۱



شکل 1. **Correlation Matrix**

Feature selection –۵–۳–۱

نتایج در فایل final.xlsx در فایل زیپ آپلود شده در سایت وجود دارد.

Supervised dataset –۶–۳–۱

در نهایت این داده‌ها با windowهای 24 ساعته یا 7 روزه آماده ورود به مدل CNN-LSTM می‌شوند. نتایج برای لگ یک روزه به صورت زیر است:

```
7 print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)
```

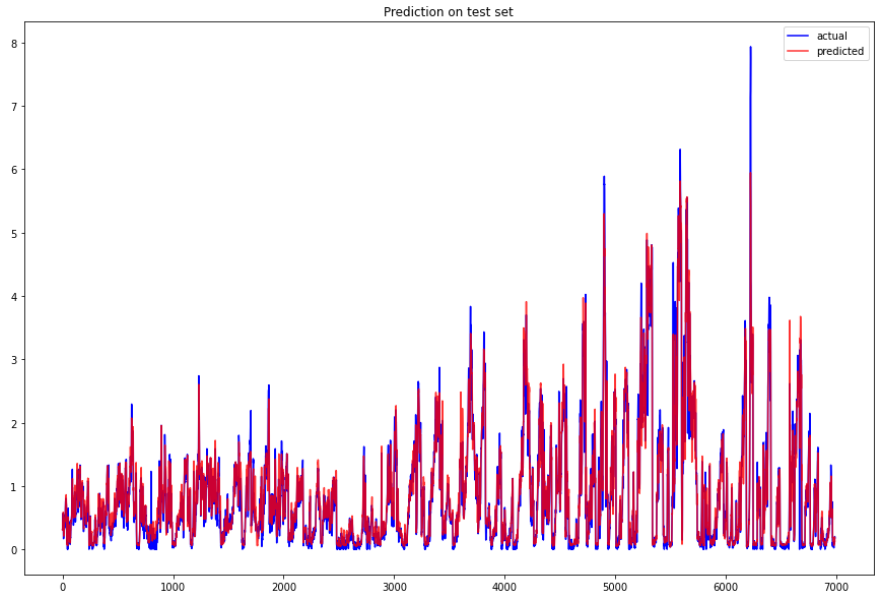
```
(28028, 24, 20) (28028,) (6988, 24, 20) (6988,)
```

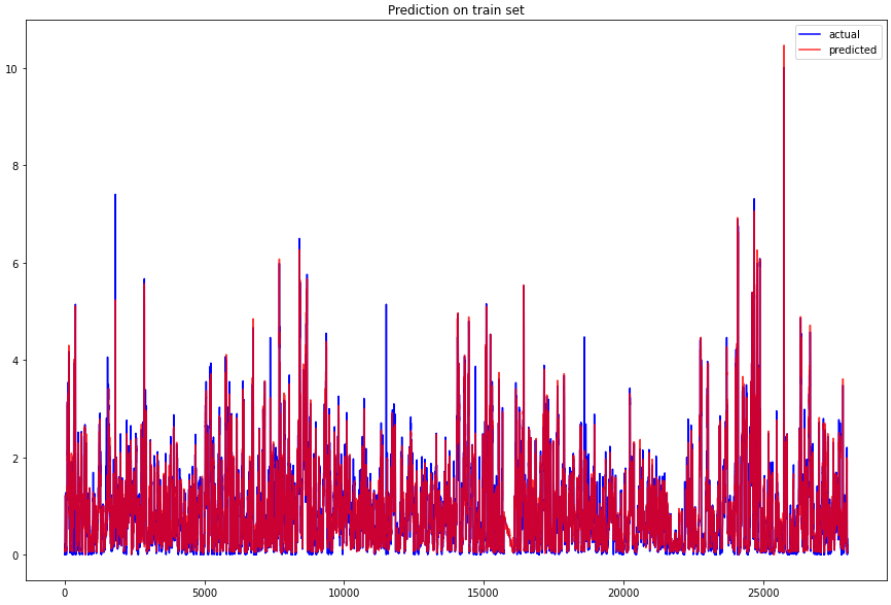
۱-۴- آموزش شبکه

به ازای Lag یک روزه:

جدول**1** . نتایج مدل برای **Lag** یک روزه

	Train	**Test**
MSE Loss	**0.0282**	**0.0414**
MAE	**0.1069**	**0.1231**
RMSE	**0.1680**	**0.2036**
R^2	**0.9725**	**0.9526**

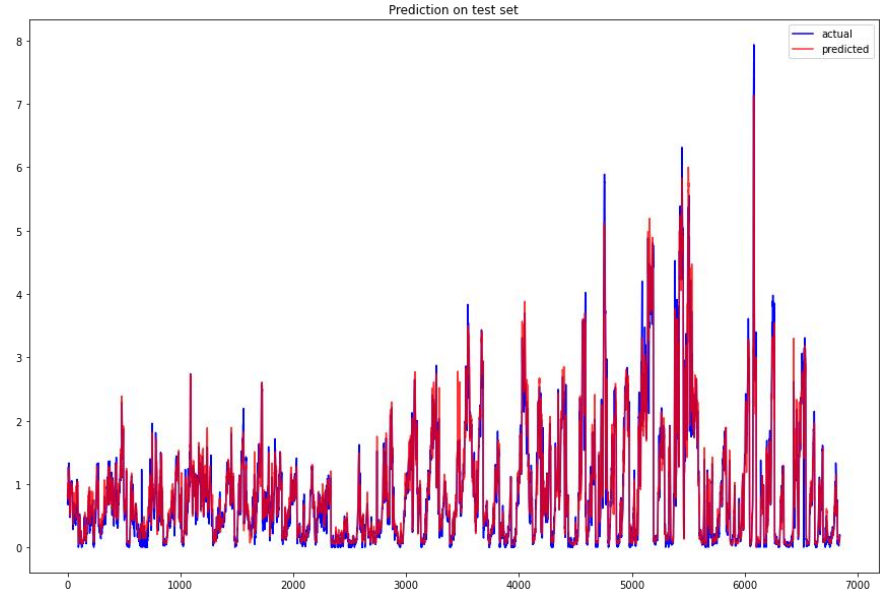


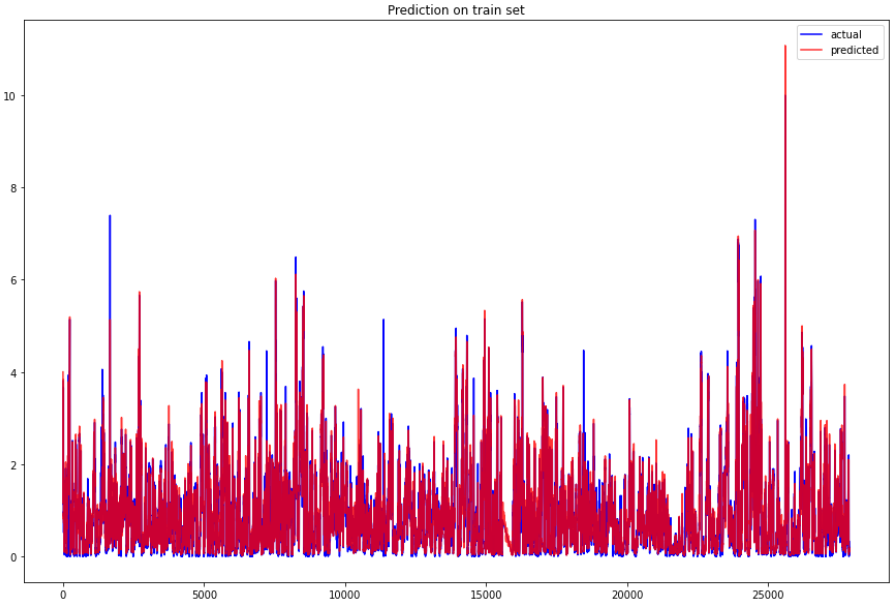


به ازای Lag هفت روزه:

جدول**2** . نتایج مدل برای **Lag** هفت روزه

	Train	**Test**
MSE Loss	**0.0313**	**0.0412**
MAE	**0.1113**	**0.1220**
RMSE	**0.1770**	**0.2029**
R^2	**0.9686**	**0.9537**

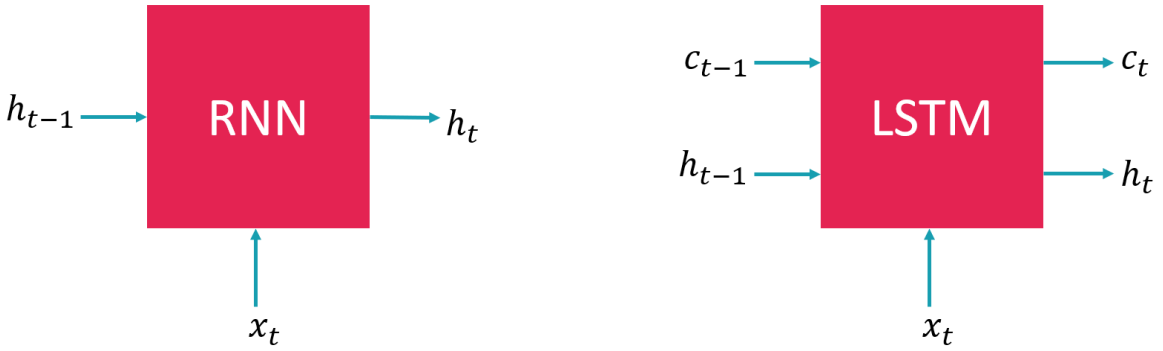




پاسخ ۲ – تشخیص اخبار جعلی

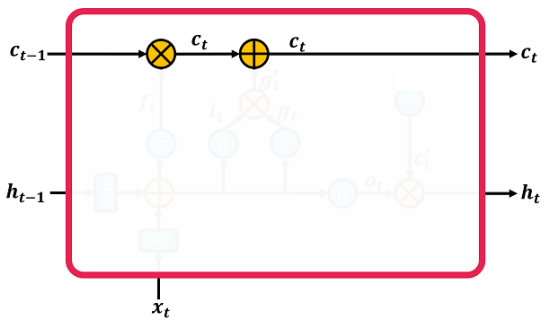
۲-۱- توضیحات مدل‌ها

شبکه‌ی عصبی LSTM یا حافظه‌ی کوتاه‌مدت طولانی (Long-Short Term Memory) نوعی خاص از شبکه عصبی بازگشتی (RNN / Recurrent Neural Network) محسوب می‌شود که مشکل حافظه‌ی بلند مدت شبکه‌ی RNN را حل می‌کند. شبکه‌ی LSTM سازوکارهایی داخلی به اسم گیت (Gate) دارد. این گیت‌ها جریان اطلاعات را کنترل می‌کنند؛ همین‌طور مشخص می‌کنند چه داده‌هایی در توالی مهم هستند و باید هم‌چنان حفظ بشوند و چه داده‌هایی باید حذف بشوند؛ به این شکل، شبکه‌ی اطلاعات مهم را در طول زنجیره‌ی توالی عبور می‌دهد تا خروجی مد نظر را داشته باشیم.



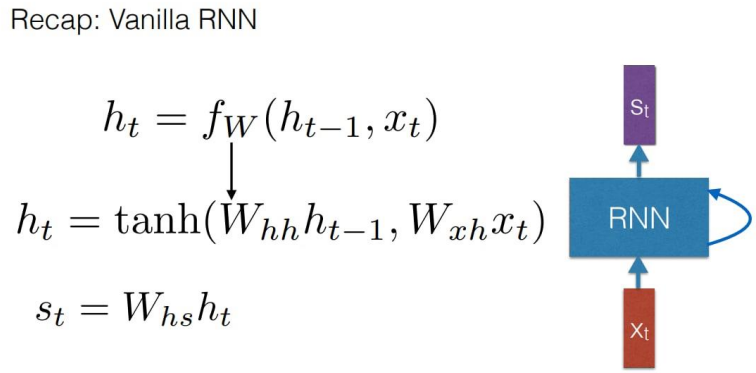
شکل 2. مقایسه معماری شبکه‌های **RNN** و **LSTM**

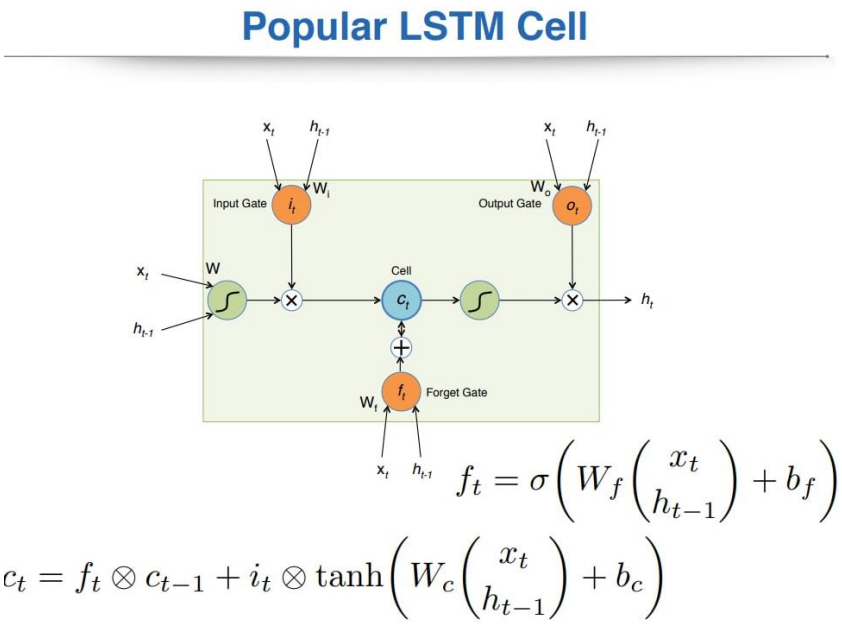
شبکه RNN یک ورودی و خروجی دارد. در واقع یک مسیر بین ورودی و خروجی شبکه RNN شکل می‌گیرد. اما شبکه LSTM متفاوت است. این شبکه دو ورودی و خروجی دارد. بین این ورودی و خروجی‌ها، یکی از ورودی‌ها مستقیم به خروجی متصل شده است! همانگونه که در شکل زیر مشخص است؛ ورودی C_{t-1} مستقیما به خروجی C_t متصل شده است. این اتصال همین‌طور ساده از اول تا آخر دنباله ادامه دارد. C مخفف Cell State هست و یک مولفه کلیدی در LSTM است. به Cell State، حافظه بلندمدت یا Long Term Memory هم گفته می‌شود.



شبکه‌های LSTM در واقع نوعی از RNNها هستند که تغییری در بلوک (RNN Unit) آن‌ها ایجاد شده است. این تغییر باعث می‌شود که شبکه‌های عصبی بازگشتیِ LSTM بتوانند مدیریت حافظه‌ی بلند مدت را داشته باشند و مشکل محوشدگی(Vannishing) یا انفجار(Explosion) گرادیان را نیز نداشته باشند.

تصاویر زیر، تفاوت معماری‌های شبکه‌ی RNN و LSTM را نمایش می‌دهند:





از شبکه‌های بازگشتی زمانی استفاده می‌شود که بین داده‌های ورودی وابستگی وجود دارد و این شبکه‌ها با توجه به ویژگی با حافظه بودنشان، می‌توانند با در نظر گرفتن این وابستگی، عملکرد بهتری نسبت به شبکه‌های Feed Forward داشته باشند. در داده‌های متنی نیز، کلمات به یکدیگر وابستگی دارند و معنی کلمه حتی می‌تواند با توجه به کلماتی در کنار آن‌ها مورد استفاده قرار می‌گیرد، تغییر کند. بنابراین استفاده از شبکه‌های بازگشتی می‌تواند این وابستگی بین داده‌های متنی را هندل کرده و عملکرد مناسبی داشته باشد.

در مدل Hybrid ارائه شده در مقاله، برخلاف شبکه‌های بازگشتی عادی، به جای اینکه داده‌ی متنی به صورت مستقیم وارد شبکه‌ی RNN یا LSTM شوند، ابتدا با استفاده از لایه‌های CNN عملیات استخراج ویژگی از داده‌ها صورت می‌گیرد و ویژگی‌های غنی (rich) محلی وارد شبکه بازگشتی می‌شوند. بدین ترتیب، هم ویژگی‌های محلی و هم وابستگی‌های Long-Term توسط LSTM، یاد گرفته می‌شوند.

۲-۲- ورودی مدل‌ها:

توضیحات Word Embeddings

Word embedding ها بردارهای عددی هستند که نمایانگر کلمات یک لغت‌نامه هستند. مفهوم اصلی word embedding این است که تمامی لغات استفاده شده در یک زبان را می‌توان توسط مجموعه ای از اعداد اعشاری (در قالب یک بردار) بیان کرد. Word embeddingها بردارهای n-بعدی‌ای هستند که تلاش می‌کنند معنای لغات و محتوای آن‌ها را با مقادیر عددی خود ثبت و ضبط کنند. هر مجموعه‌ای از اعداد یک "بردار کلمه" معتبر به حساب می‌آید که الزاما برای ما سودمند نیست، آن مجموعه ای از بردار کلمات برای کاربردهای مورد نظر ما سودمندند که معنای کلمات ، ارتباط بین آنها و محتوای کلمات مختلف را همانطور که به صورت طبیعی مورد استفاده قرار گرفته‌اند، بدست آورده باشند.

در سال 2013، Word embeddings به عنوان روش جدیدی برای vectorize کردن متن، ارائه شد و باعث انقلاب بزرگی در حوزه NLP شد. در فاصله‌های زمانی خیلی کوتاه سه تکنیک embedding با نام‌های Word2vec، GloVe و fastText معرفی شدند.

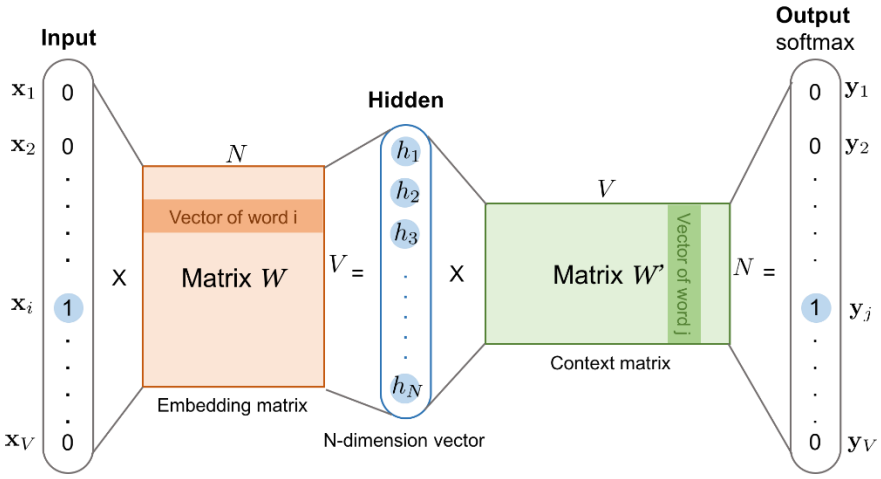
- سال 2013: تکنیک word2vec توسط Thomas Mikolov در Google ارائه شد.
- سال 2014: تکنیک GloVe توسط Jeffrey Pennington در Stanford ارائه شد.
- سال 2016: تکنیک fastText توسط Piotr Bojanowski در Facebook ارائه شد.

متدهای Word embedding به عنوان روش‌های جدید بردارسازی متن، کاستی‌ها و معایب ذاتی tf-idf را حل می‌کنند. در این تکنیک‌ها، شباهت معنایی (semantic similarity) بین کلمات حفظ می‌شود؛ به عبارتی دیگر، با بردارهای به‌دست‌آمده از این تکنیک‌ها، می‌توان معنای کلمات را تشخیص داد و میزان شباهت کلمات مختلف را با یکدیگر به دست آورد.

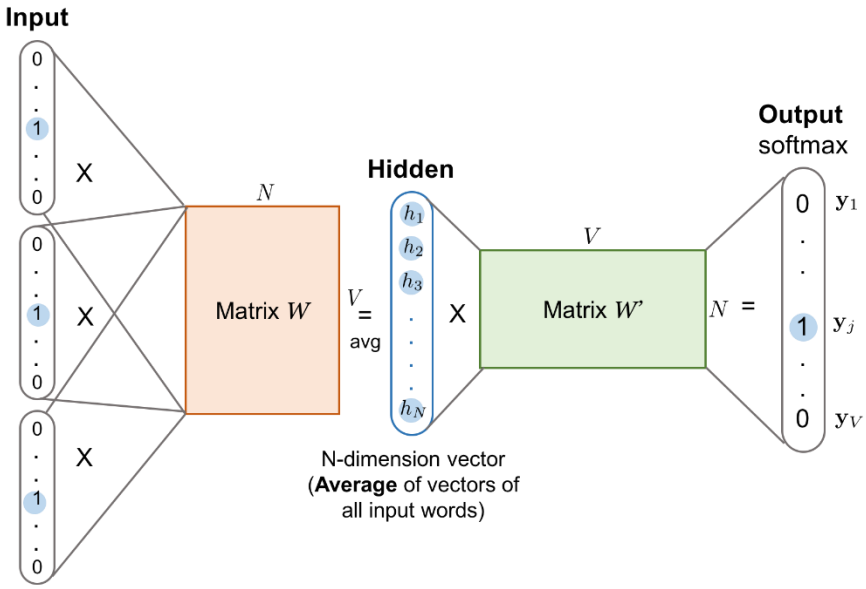
دو روش اصلی در رابطه با یادگیری word embedding وجود دارد که هر دوی آنها وابسته به دانش محتوایی‌اند:

- مبتنی بر شمارش (Count-based)
- مبتنی بر محتوا

از جمله روش‌های مبتنی بر محتوا می‌توان به Skip-Gram و Continuous Bag of Words اشاره کرد



شکل 3. **Skip-gram**



شکل 4. **Continuous Bag of Words**

روش استفاده شده برای پیاده‌سازی این شبکه، تکنیک GloVe می‌باشد.

۲-۳- پیاده‌سازی

۲-۳-۱- پیش پردازش

پیش‌پردازش‌های لازم:

1. حذف کردن URLها و Stop Words:

```
[ ]  1 df['article_content'].str.contains('https').sum()

     14

[ ]  1 df['article_content'] = df['article_content'].str.replace(r'http\S+', '', regex=True)
     2 df['article_content'].str.contains('https').sum()

     0

[ ]  1 stop = stopwords.words('english')
     2 df['article_content'] = df['article_content'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
     3 df.head()
```

	unit_id	article_title	article_content	source	date	location	labels
0	1914947530	Syria attack symptoms consistent with nerve ag...	Wed 05 Apr 2017 Syria attack symptoms consiste...	nna	4/5/2017	idlib	0
1	1914947532	Homs governor says U.S. attack caused deaths b...	Fri 07 Apr 2017 0914 Homs governor says U.S. a...	nna	4/7/2017	homs	0
2	1914947533	Death toll from Aleppo bomb attack at least 112	Sun 16 Apr 2017 Death toll Aleppo bomb attack ...	nna	4/16/2017	aleppo	0
3	1914947534	Aleppo bomb blast kills six Syrian state TV	Wed 19 Apr 2017 Aleppo bomb blast kills six Sy...	nna	4/19/2017	aleppo	0
4	1914947535	29 Syria Rebels Dead in Fighting for Key Alepp...	Sun 10 Jul 2016 29 Syria Rebels Dead Fighting ...	nna	7/10/2016	aleppo	0

2. استفاده از Stemmer:

```
[ ]  1 stemmer = SnowballStemmer("english")
     2
     3 df['article_content'] = df['article_content'].apply(
     4         lambda x: ' '.join([stemmer.stem(y) for y in x.split()])
     5     )
     6 df.head()
```

	unit_id	article_title	article_content	source	date	location	labels
0	1914947530	Syria attack symptoms consistent with nerve ag...	wed 05 apr 2017 syria attack symptom consist n...	nna	4/5/2017	idlib	0
1	1914947532	Homs governor says U.S. attack caused deaths b...	fri 07 apr 2017 0914 hom governor say u.s. att...	nna	4/7/2017	homs	0
2	1914947533	Death toll from Aleppo bomb attack at least 112	sun 16 apr 2017 death toll aleppo bomb attack ...	nna	4/16/2017	aleppo	0
3	1914947534	Aleppo bomb blast kills six Syrian state TV	wed 19 apr 2017 aleppo bomb blast kill six syr...	nna	4/19/2017	aleppo	0
4	1914947535	29 Syria Rebels Dead in Fighting for Key Alepp...	sun 10 jul 2016 29 syria rebel dead fight key ...	nna	7/10/2016	aleppo	0

3. حذف کردن کلمات و عبارات مربوط به تاریخ:

```
[ ]  1 days = ['sat', 'sun', 'mon', 'tue', 'wed', 'thu', 'fri']
     2 months = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec']
     3 date_words = [str(i) for i in range(1970, 2023)] + days + [str(i).zfill(2) for i in range(1, 32)] + months
     4
     5 df['article_content'] = df['article_content'].apply(lambda x: ' '.join([y for y in x.split() if y not in date_words]))
     6 df.head()
```

	unit_id	article_title	article_content	source	date	location	labels
0	1914947530	Syria attack symptoms consistent with nerve ag...	syria attack symptom consist nerv agent use wh...	nna	4/5/2017	idlib	0
1	1914947532	Homs governor says U.S. attack caused deaths b...	0914 hom governor say u.s. attack caus death d...	nna	4/7/2017	homs	0
2	1914947533	Death toll from Aleppo bomb attack at least 112	death toll aleppo bomb attack least 112. the d...	nna	4/16/2017	aleppo	0
3	1914947534	Aleppo bomb blast kills six Syrian state TV	aleppo bomb blast kill six syrian state tv. a ...	nna	4/19/2017	aleppo	0
4	1914947535	29 Syria Rebels Dead in Fighting for Key Alepp...	syria rebel dead fight key aleppo road. at lea...	nna	7/10/2016	aleppo	0

4. استفاده از Tokenizer:

```
[ ]  1 t_train = Tokenizer()
     2 print(x_train[0])
     3 t_train.fit_on_texts(df['article_content'])
     4 x_train = t_train.texts_to_sequences(x_train)
     5 x_train = tf.keras.preprocessing.sequence.pad_sequences(x_train, maxlen=100, padding='post')
     6 print(x_train[0])
```

```
25-12-2014 isil milit kill clash kurd ne syria. at least milit terrorist group oper iraq syria kill clash kurdish troop northeastern syria
[  24    2  104   20   61    9   11  100  145    6    4  173   81  201
  408    6  122   24    2    1   66 1323  107   61  147   81   10  462
   73  357   62 8078  501   32   67    3   18   45  602    1  357 1067
   49   62   35   94  252    9    2  169  109   11    1  520   34  735
   68   46  173  333  107  139   85  353  495  168   80   17  233  145
    6  312  410  434 1145  529  621   80   13   29  682   21   79   81
   46  750   13    6   21   44  353 1483 1072    7  788 1067   17  252
  342  511]
```

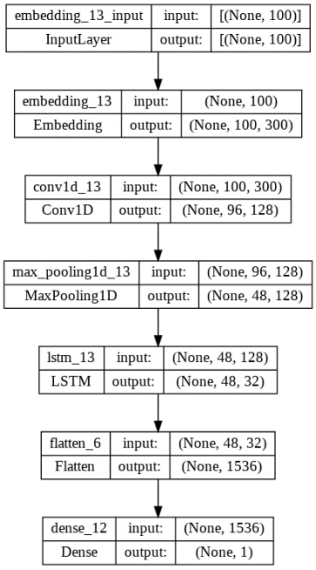
5. استفاده از GloVe:

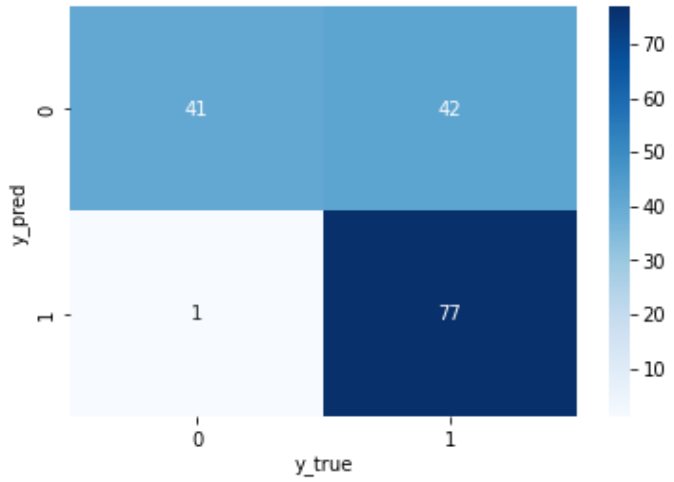
```
[ ]  1 embeddings_index = dict()
     2 f = open('glove.6B.300d.txt')
     3 for line in f:
     4     values = line.split()
     5     word = values[0]
     6     coefs = np.asarray(values[1:], dtype='float32')
     7     embeddings_index[word] = coefs
     8 f.close()
```

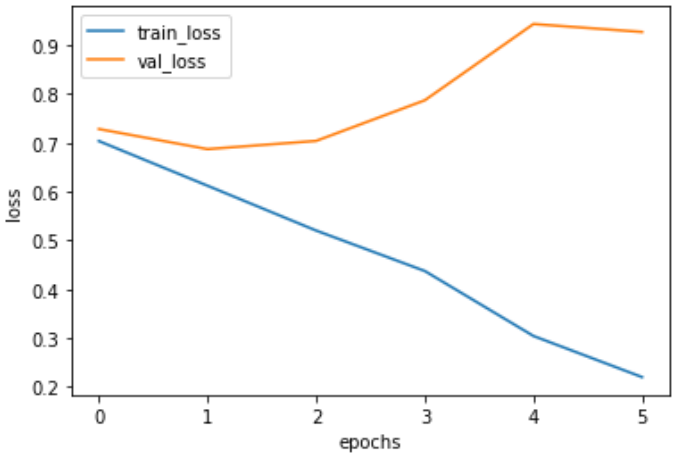
```
[ ]  1 embedding_matrix = np.zeros((len(t_train.word_index)+1, 300))
     2 for word, i in t_train.word_index.items():
     3     embedding_vector = embeddings_index.get(word)
     4     if embedding_vector is not None:
     5         embedding_matrix[i] = embedding_vector
```

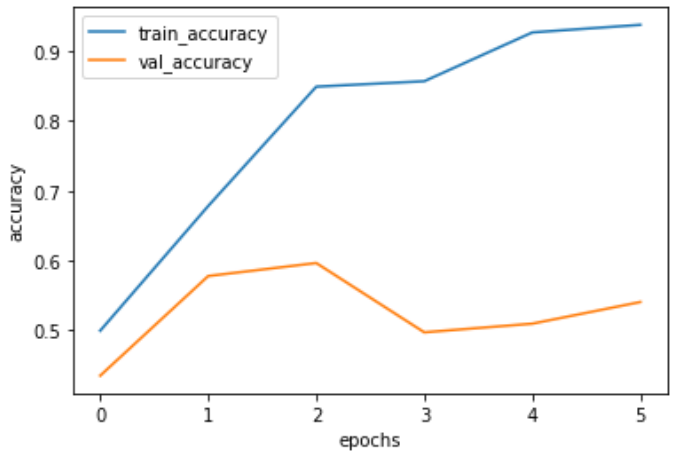
۲-۳-۲- آموزش مدل‌ها

مدل hybrid (CNN-RNN):





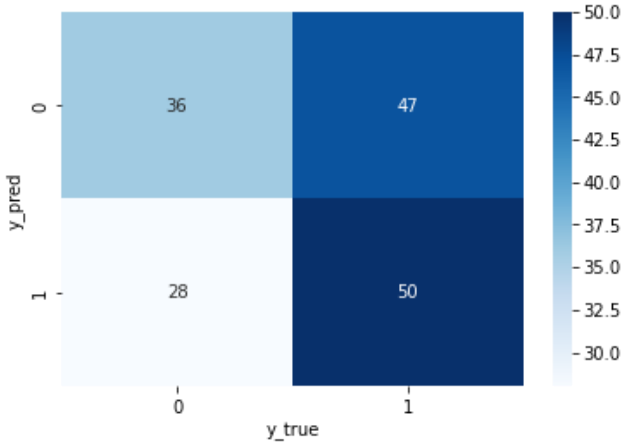


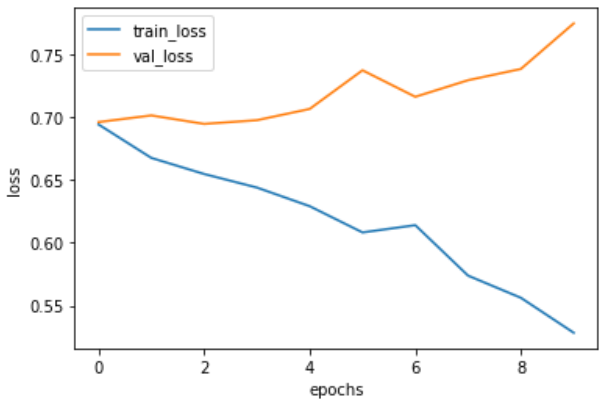


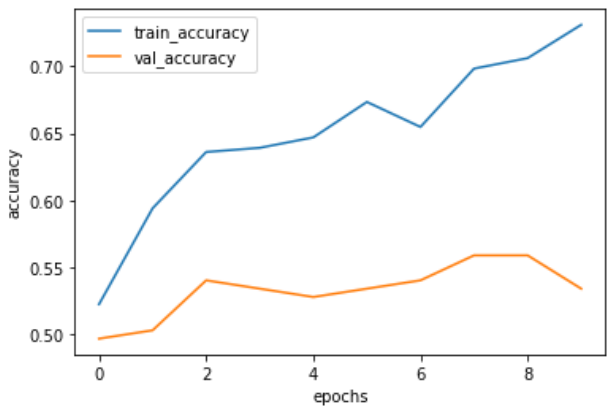
	precision	recall	f1-score	support
0	0.98	0.49	0.66	83
1	0.65	0.99	0.78	78
accuracy			0.73	161
macro avg	0.81	0.74	0.72	161
weighted avg	0.82	0.73	0.72	161

مدل RNN:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 300)          2540700

 lstm_1 (LSTM)               (None, 64)                93440

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 2,634,205
Trainable params: 93,505
Non-trainable params: 2,540,700
_________________________________________________________________
```







	precision	recall	f1-score	support
0	0.56	0.43	0.49	83
1	0.52	0.64	0.57	78
accuracy			0.53	161
macro avg	0.54	0.54	0.53	161
weighted avg	0.54	0.53	0.53	161

همانطور که از نتایج فوق مشخص است، مدل هیبرید نتایج بهتری نسبت به مدل RNN ساده دارد. دلیل این موضوع هم طبق توضیحات ارائه شده در بالا، ترکیب استخراج ویژگی محلی و long-term در مدل هیبرید است که غنای ویژگی‌ها را بیشتر کرده و عملیات طبقه‌بندی را بهبود می‌بخشد.

۲-۴- تحلیل نتایج

ایده‌ای که برای بهبود مدل ارائه شده در مقاله می‌توان استفاده کرد این است که به جای استفاده از آخرین hidden state موجود در LSTM از کل hidden state ها استفاده کرده و در واقع مانند مدل Attention عمل کنیم. این موضوع دقت را از 60 درصد به حدود 73 درصد در مدل هیبرید افزایش داده است.