



دانشگاه شهید بهشتی

دانشکده علوم ریاضی

Text Classification

تمرین چهارم درس شبکه عصبی

نگارش

زهرا موسی خانی

استاد راهنما

دکتر سعیدرضا خردپیشه

بهار ۱۴۰۰

فهرست مطالب

۱	دیتاست	۱
۳	<i>pre processing</i>	۲
۸	<i>word embedding</i>	۳
۱۰	<i>text classifier</i>	۴

فهرست شکل‌ها

۱.۱	شکل اول : نمونه داده ها	۲
۱.۲	شکل دوم : کوچک کردن تمام حروف متن	۳
۲.۲	شکل سوم : مخف شده ها	۴
۳.۲	شکل چهارم : حذف لینک ها	۴
۴.۲	شکل پنجم : معادل سازی کلمات مخفف	۴
۵.۲	شکل ششم : توکن کردن متن ها	۵
۶.۲	شکل هفتم : حذف <i>stop words</i>	۵
۷.۲	شکل هشتم : <i>stemming</i>	۶
۸.۲	شکل نهم : <i>POS</i>	۷
۹.۲	شکل دهم : <i>lemmatization</i>	۷
۱.۳	شکل یازدهم : <i>word embedding</i>	۹
۲.۳	شکل دوازدهم : ساخت وکب	۹
۱.۴	شکل سیزدهم : <i>LSTM</i>	۱۰
۲.۴	شکل چهاردهم : مدل اول	۱۱
۳.۴	شکل پانزدهم : مدل دوم	۱۱
۴.۴	شکل شانزدهم : مدل سوم	۱۲

۵.۴ شکل هفدهم: مدل چهارم ۱۲

فصل ۱

دیتاست

دیتاست این تمرین شامل توییت هایی در خصوص وقایع طبیعی می باشد. هدف این تمرین این است که بیان کنیم که یک توییت در خصوص یک واقعه طبیعی واقعی است یا سک واقعه طبیعی جعلی. در واقع هدف ایجاد یک *classifier* می باشد که تعیین کند توییت ها متعلق به کدام یک از دو کلاس گفته شده می باشند. ابتدا داده هارا از روی فایل *csv* میخوانیم و دیتاست خود را از روی فایل درست میکنیم. در شکل شماره ۱ نحوه انجام این کار و نمونه ای از دیتا ایجاد شده را میتوانیم ببینیم.



```
#read the data:
train_df = pd.read_csv("/content/train.csv")
display(train_df.shape, train_df.head())
```

(7613, 5)

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

```
[ ] #some early exploration:
display(train_df[~train_df["location"].isnull()].head())
display(train_df[train_df["target"] == 0]["text"].values[1])
display(train_df[train_df["target"] == 1]["text"].values[1])
```

	id	keyword	location	text	target
31	48	ablaze	Birmingham	@bbcmtd Wholesale Markets ablaze http://t.co/l...	1
32	49	ablaze	Est. September 2012 - Bristol	We always try to bring the heavy. #metal #RT h...	0
33	50	ablaze	AFRICA	#AFRICANBAZE: Breaking news:Nigeria flag set a...	1
34	52	ablaze	Philadelphia, PA	Crying out for more! Set me ablaze	0
35	53	ablaze	London, UK	On plus side LOOK AT THE SKY LAST NIGHT IT WAS...	0

شکل ۱.۱: شکل اول : نمونه داده ها

فصل ۲

pre processing

درهنگام کار با داده های متنی، نیاز به یک سری پیش پردازش هایی است تا روی متن اعمال شوند پیش از آنکه داده ها به مدل داده شوند. در این بخش به این پیش پردازش ها میپردازیم. در مرحله اول تمام حروف را به حروف کوچک تبدیل کردیم و در ستون جدیدی متن های جدید را ذخیره نمودیم. در شکل شماره ۲ نتیجه و چگونگی انجام این کار را مشاهده میکنید.

در مرحله بعد از پکیج *contraction* برای اینکه کلمات مخفف شده را به حالت عادی برگردانیم استفاده کردیم. یک مثال از این نمونه می تواند این باشد که *I'll* را به *I will* تبدیل کنیم. نتایج این تبدیل در شکل شماره ۳ قابل مشاهده می باشد.

یک متن ممکن است شامل کاراکترهای غیر ضروری ای مانند *URL* ها و تگ های *HTML* و غیره باشد. در

```
[ ] train_df["text_clean"] = train_df["text"].apply(lambda x: x.lower())
display(train_df.head())
```

	id	keyword	location	text	target	text_clean
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this #earthquake m...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask. canada
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to 'shelter in place' are ...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13,000 people receive #wildfires evacuation or...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby #alaska as ...

شکل ۱.۲: شکل دوم: کوچک کردن تمام حروف متن

```
'I can't have kids cuz I got in a bicycle accident & split my testicles. it's impossible for me to have kids' MICHAEL YOU ARE THE FATHER
'i cannot have kids cuz i got in a bicycle accident & split my testicles. it is impossible for me to have kids' michael you are the father
#raining #flooding #Florida #TampaBay #Tampa 18 or 19 days. I've lost count
#raining #flooding #florida #tampabay #tampa 18 or 19 days. I have lost count
```

شکل ۲.۲: شکل سوم: مخف شده ها

```
@bbcmtd Wholesale Markets ablaze http://t.co/1HYXE0HY6C
@bbcmtd wholesale markets ablaze
INEC Office in Abia Set Ablaze - http://t.co/3ImaomknnA
inec office in abia set ablaze -
Rene Ablaze & Jacinta - Secret 2k13 (Fallen Skies Edit) - Mar 30 2013 https://t.co/7MLMsUzV1Z
rene ablaze & jacinta - secret 2k13 (fallen skies edit) - mar 30 2013
```

شکل ۳.۲: شکل چهارم: حذف لینک ها

این مرحله با نوشتن توابعی برای حذف این دسته از کاراکترهای اضافی متن، متن را از این کاراکترها پاک میکنیم. توابع نوشته شده برای این منظور در کد قابل مشاهده می باشند بنابراین از آوردن مجدد آنها در این بخش خودداری میکنیم و صرفا از آوردن یک نمونه که به حذف لینک های موجود در متن میپردازد در شکل ۴ بسنده میکنیم. به همین ترتیب به حذف پانکچویشن ها نیز پرداختیم.

در مرحله بعد یک تابعی را آماده کردیم از مخفف کلمات. همان طور که میدانیم در متن هایی مانند توییت ممکن است بعضی کلمات به صورت مخفف نوشته شده باشند. با استفاده از این تابع و مپ کردن این کلمات مخفف به معادل اصلی شان مرحله بعدی را طی کردیم. نمونه ای از این تبدیل ه را در شکل شماره ۵ می توان دید.

حال در این مرحله به توکن کردن متن ها میپردازیم و نمونه این کار را در شکل شماره ۶ مشاهده میکنید. در این مرحله، به حذف *stop word* ها میپردازیم. این کلمات، کلماتی هستند که به فراوانی زیادی در یک متن ظاهر میشوند. برای مثال *a* , *and* , *the* , ما با استفاده از *nlTK.corpus* ابتدا مجموعه ای از این کلمات را در زبان انگلیسی دانلود کردیم. سپس به حذف آنها پرداختیم. روند کار و نتایج را میتوان در شکل ۷ مشاهده کرد.

```
MH370: Intact part lifts odds plane glided not crashed into sea http://t.co/8pdmHH6tzH
malaysia airlines flight 370 intact part lifts odds plane glided not crashed into sea
@USAgov Koreans are performing hijacking of the Tokyo Olympic Games. https://t.co/APkSnpLXZj
usa government koreans are performing hijacking of the tokyo olympic games
```

شکل ۴.۲: شکل پنجم: معادل سازی کلمات مخفف


```
# Tokenizing the tweet base texts.
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

train_df["tokenized"] = train_df["text_clean"].apply(word_tokenize)
train_df.head()
```

[nltk_data] Downloading package punkt to /root/nltk_data...

[nltk_data] Unzipping tokenizers/punkt.zip.

	id	keyword	location	text	target	text_clean	tokenized
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...

شکل ۵.۲: توکن‌ششم: توکن کردن متن‌ها

```
[ ] # Removing stopwords.
nltk.download("stopwords")
from nltk.corpus import stopwords

stop = set(stopwords.words('english'))
train_df['stopwords_removed'] = train_df['tokenized'].apply(lambda x: [word for word in x if word not in stop])
train_df.head()
```

[nltk_data] Downloading package stopwords to /root/nltk_data...

[nltk_data] Unzipping corpora/stopwords.zip.

	id	keyword	location	text	target	text_clean	tokenized	stopwords_removed
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	[13000, people, receive, wildfires, evacuation...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...	[got, sent, photo, ruby, alaska, smoke, wildfi...

شکل ۶.۲: حذف *stop words*: حذف

Stemming یک پروسه استخراج ریشه کلمات می باشد. به این معنا که از بین شکل های مختلفی که یک کلمه میتواند داشته باشد (به عنوان مثال شکل جمع، مفرد، فعل و ..) ریشه آن کلمه را استخراج میکند. سه الگوریتم معروف برای این منظور *portet* و *snowball* و *Lancaster* می باشند و با توجه به مقبولیت بیشتری که الگوریتم دوم دارد، از این الگوریتم برای این منظور استفاده میکنیم و نمونه ای از نتایج و شیوه استفاده از این الگوریتم را در شکل شماره ۸ مشاهده میکنیم.

part of specech tagging: در این مرحله نقش کلمات در جمله را با استفاده از سه روش معروف *pos tag* و *UnigramTagger* و *BigramTagger* روش استفاده شده به طور کامل در کد قابل مشاهده می باشد

```
[31] #stemming:
from nltk.stem import SnowballStemmer

def snowball_stemmer(text):
    """
    Stem words in list of tokenized words with SnowballStemmer
    """
    stemmer = nltk.SnowballStemmer("english")
    stems = [stemmer.stem(i) for i in text]
    return stems

train_df['snowball_stemmer'] = train_df['stopwords_removed'].apply(lambda x: snowball_stemmer(x))
train_df.head()
```

	id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	snowball_stemmer
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[deed, reason, earthquak, may, allah, forgiv, us]
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...	[resid, ask, shelter, place, notifi, offic, ev...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	[13000, people, receive, wildfires, evacuation...	[13000, peopl, receiv, wildfir, evacu, order, ...]
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...	[got, sent, photo, ruby, alaska, smoke, wildfi...	[got, sent, photo, rubi, alaska, smoke, wildfi...

شکل ۷.۲: شکل هشتم: *stemming*

و در اینجا صرفاً یک نمونه از خروجی را در شکل ۹ قرار میدهم. همان طور که در ستون آخر دیتاست مشاهده میکنیم، کلمات را به n و افعال را به v مپ کرده است.

Lemmatization

پروسه ای است که تشخیص میدهد دو کلمه هر چند تفاوت های ظاهری داشته باشند، داریم ریشه یکسانی اند یا درواقع میتوان گفت در یک دسته قرار میگیرند. برای مثال کلمات *am* , *is* , *are* یک لم دارند و درواقع فعل *to be* می باشند. برای این منظور در این تمرین از *WordNetLemmatizer* استفاده کردیم. نتیجه را در شکل ۱۰ مشاهده میکنیم.

```
[ ] train_df['combined_postag_wnet'] = train_df['stopwords_removed'].apply(lambda x: pos_tag_wordnet(x))
train_df.head()
```

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	snowball_stemmer	combined_postag_wnet	
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[deed, reason, earthquak, may, allah, forgiv, us]	[(deeds, n), (reason, n), (earthquake, n), (ma...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, rong, sask, canada]	[(forest, n), (fire, n), (near, n), (la, n), (...]
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...	[resid, ask, shelter, place, notifi, offic, ev...	[(residents, n), (asked, v), (shelter, n), (pl...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	[13000, people, receive, wildfires, evacuation...	[13000, peopl, receiv, wildfir, evacu, order, ...]	[(13000, n), (people, n), (receive, v), (wildf...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...	[got, sent, photo, ruby, alaska, smoke, wildfi...	[got, sent, photo, rubi, alaska, smoke, wildfi...	[(got, v), (sent, v), (photo, n), (ruby, n), (...]

شکل ۸.۲: POS

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	snowball_stemmer	combined_postag_wnet	lemmatize_word_w_pos	lemmatize_text
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[deed, reason, earthquak, may, allah, forgiv, us]	[(deeds, n), (reason, n), (earthquake, n), (ma...	deed reason earthquake may allah forgive u
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, rong, sask, canada]	[(forest, n), (fire, n), (near, n), (la, n), (...]	forest fire near la ronge sask canada
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...	[resid, ask, shelter, place, notifi, offic, ev...	[(residents, n), (asked, v), (shelter, n), (pl...	resident ask shelter place notified officer ev...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	[13000, people, receive, wildfires, evacuation...	[13000, peopl, receiv, wildfir, evacu, order, ...]	[(13000, n), (people, n), (receive, v), (wildf...	13000 people receive wildfire evacuation order...
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...	[got, sent, photo, ruby, alaska, smoke, wildfi...	[got, sent, photo, rubi, alaska, smoke, wildfi...	[(got, v), (sent, v), (photo, n), (ruby, n), (...]	get send photo ruby alaska smoke wildfire pour...

شکل ۹.۲: lemmatization

فصل ۳

word embedding

در هنگام کار با داده های متنی روش هایی وجود دارد برای تبدیل متن ها به وکتور هایی از اعداد که این اعداد سعی در استخراج ویژگی های معنایی، ارتباط کلمات و غیره را دارند. و درواقع یک نمایش عددی از متن های ما میباشد که با این اعداد ویژگی های استخراج شده از متن را نشان میدهند. روش های بسیاری برای این امر وجود دارد که از جمله معروف ترین آنها *Word2Vec* میباشد. در جدولی که در شکل شماره ۱۱ آمده است مقایسه و مروری بر این روش ها مشاهده میکنیم. ما در این تمرین از روش *glove* استفاده کردیم. در هنگام ساخت یک دیکشنری از کلمات موجود با استفاده از *Text.build vocab* پارامتر *vectors* را برابر *glove* انتخاب کردیم و در شکل ۱۲ نیز کد آن را مشاهده میکنیم.

Model	Advantages	Limitation
Weighted Words	<i>Easy to compute</i> Easy to compute the similarity between 2 documents using it Basic metric to extract the most descriptive terms in a document Works with an unknown word (e.g., New words in languages)	<i>It does not capture the position in the text (syntactic)</i> It does not capture meaning in the text (semantics) * Common words effect on the results (e.g., "am", "is", etc.)
TF-IDF	<i>Easy to compute</i> Easy to compute the similarity between 2 documents using it Basic metric to extract the most descriptive terms in a document Common words do not affect the results due to IDF (e.g., "am", "is", etc.)	<i>It does not capture the position in the text (syntactic)</i> It does not capture meaning in the text (semantics)
Word2Vec	<i>It captures the position of the words in the text (syntactic)</i> It captures meaning in the words (semantics)	<i>It cannot capture the meaning of the word from the text (fails to capture polysemy)</i> It cannot capture out-of-vocabulary words from corpus
GloVe (Pre-Trained)	<i>It captures the position of the words in the text (syntactic)</i> It captures meaning in the words (semantics) * Trained on huge corpus	<i>It cannot capture the meaning of the word from the text (fails to capture polysemy)</i> Memory consumption for storage * It cannot capture out-of-vocabulary words from corpus
GloVe (Trained)	<i>It is very straightforward, e.g., to enforce the word vectors to capture sub-linear relationships in the vector space (performs better than Word2vec)</i> Lower weight for highly frequent word pairs, such as stop words like "am", "is", etc. Will not dominate training progress	Memory consumption for storage Needs huge corpus to learn <i>It cannot capture out-of-vocabulary words from the corpus</i> It cannot capture the meaning of the word from the text (fails to capture polysemy)
FastText	<i>Works for rare words (rare in their character n-grams which are still shared with other words)</i> Solves out of vocabulary words with n-gram in character level	<i>It cannot capture the meaning of the word from the text (fails to capture polysemy)</i> Memory consumption for storage * Computationally is more expensive in comparing with GloVe and Word2Vec
Contextualized Word Representations	* It captures the meaning of the word from the text (incorporates context, handling polysemy)	Memory consumption for storage Improves performance notably on downstream tasks. Computationally is more expensive in comparison to others Needs another word embedding for all LSTM and feedforward layers It cannot capture out-of-vocabulary words from a corpus * Works only sentence and document level (it cannot work for individual word level)

شکل ۱.۳: شکل یازدهم: *word embedding*

```
[ ] from collections import Counter
MAX_VOCAB_SIZE = 25000
#vocab = Counter(train_ds[:]['lemmatize_text']) # create a dictionary
#vocab = sorted(vocab, key=vocab.get, reverse=True)
#MAX_VOCAB_SIZE = len(vocab)
print(MAX_VOCAB_SIZE)
TEXT.build_vocab(train_ds,
                  max_size = MAX_VOCAB_SIZE,
                  vectors = 'glove.6B.200d',
                  unk_init = torch.Tensor.zero_)
```

شکل ۲.۳: شکل دوازدهم: ساخت وکب

فصل ۴

text classifier

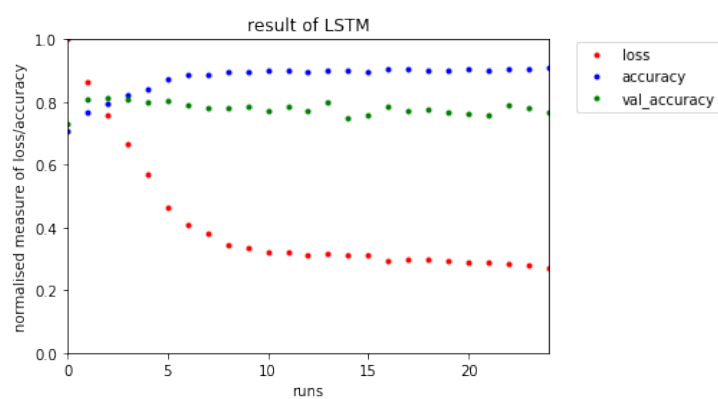
ما برای طبقه بندی توییت ها در این تمرین از *LSTM* استفاده کردیم که هایپارامترها و ساختار به طور جزئی در کد موجود میباشد و در شکل ۱۳ خلاصه ای از ساختار *LSTM* را مشاهده میکنیم.

حال به بررسی نتایج و آپتیمایزرهای مختلف و تاثیر آنها بر دقت مدل میپردازیم. ابتدا نتایج حاصل با استفاده از آپتیمایزر *Adam* و تابع خطا *BCEWithLogitsLoss* با لرنینگ ریت یک هزارم و به تعداد ۲۵ اپاک را در شکل شماره ۱۴ مشاهده میکنیم. در مرحله بعد، تعداد اپاک ها را به ۱۰۰ اپاک افزایش داده اما نتایج چندان تغییری نکرد. در شکل ۱۵ روند کاهش خطا و افزایش دقت مدل را برای ۱۰۰ اپاک مشاهده میکنیم.

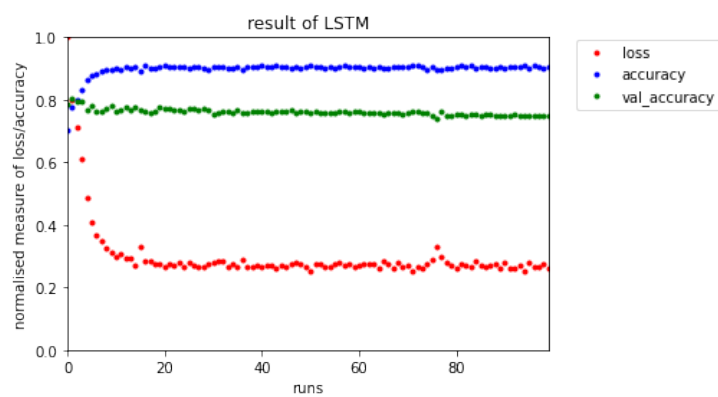
در مرحله بعد از آپتیمایزر *SGD* استفاده کردیم با لرنینگ ریت ۰.۰۱ و *momentum* برابر نه دهم به تعداد ۱۰۰ اپاک که همان طور که در شکل ۱۶ مشاهده میکنیم نتایج خیلی مطلوبی حاصل نشد و نهایتاً به *accuracy* ۷۴ درصد میرسیم این درحالیست که در مدل قبل به ۹۴ درصد رسیده بودیم. برای اینکه نتایج بهتری با این

```
LSTM_net(  
    (embedding): Embedding(12931, 200, padding_idx=1)  
    (rnn): LSTM(200, 256, num_layers=2, dropout=0.2, bidirectional=True)  
    (fc1): Linear(in_features=512, out_features=256, bias=True)  
    (fc2): Linear(in_features=256, out_features=1, bias=True)  
    (dropout): Dropout(p=0.2, inplace=False)  
)
```

شکل ۱.۴: شکل سیزدهم: *LSTM*

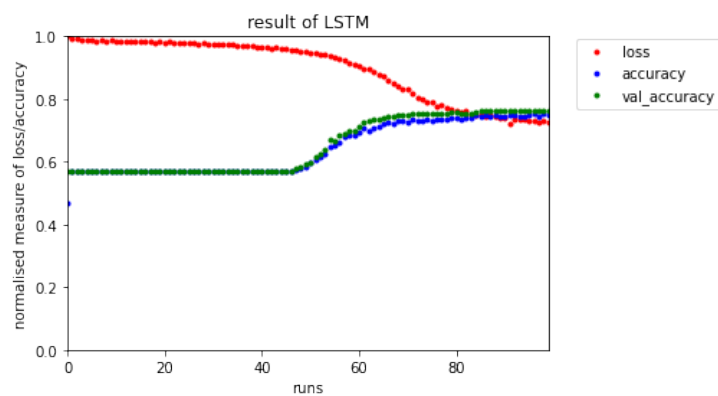


شکل ۲.۴: شکل چهاردهم: مدل اول

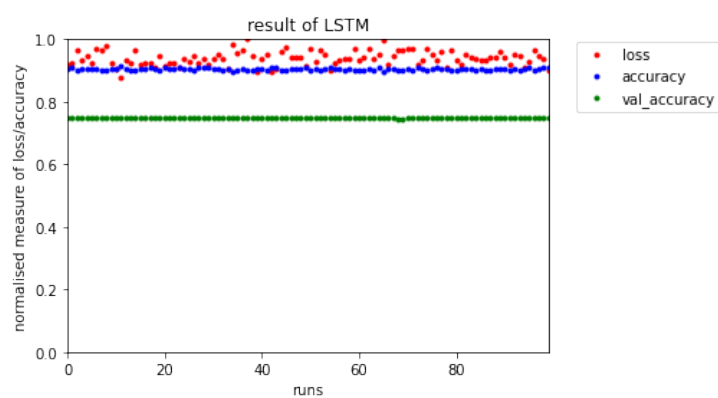


شکل ۳.۴: شکل پانزدهم: مدل دوم

آپتیمایزر بگیریم لرنینگ ریت را به یک صدم و $momentum$ را به نیم تغییر دادیم و نتایج را در شکل ۱۷ مشاهده میکنیم.



شکل ۴.۴: شکل شانزدهم: مدل سوم



شکل ۵.۴: شکل هفدهم: مدل چهارم