

به نام خدا



تمرین سری دوازدهم درس یادگیری عمیق

دکتر محمدی

محمد یارمقدم

۹۶۴۶۲۱۰۴

سوال اول)

در این سوال هدف کار بر روی دیتاست مربوط به تاریخچه قیمت های بیت کوین است.

برای این هدف، ابتدا از کتابخانه **yfinance** دیتاست را در بازه مشخص شده دانلود و سپس از روی آرگومان **close** داده های موجود نمودار قیمت آن را در بازه مورد نظر رسم کردم. سپس مقادیر این داده ها را در آرایه جمع آوری و **shape** آن را به حالت مورد نظر درمیاوریم.

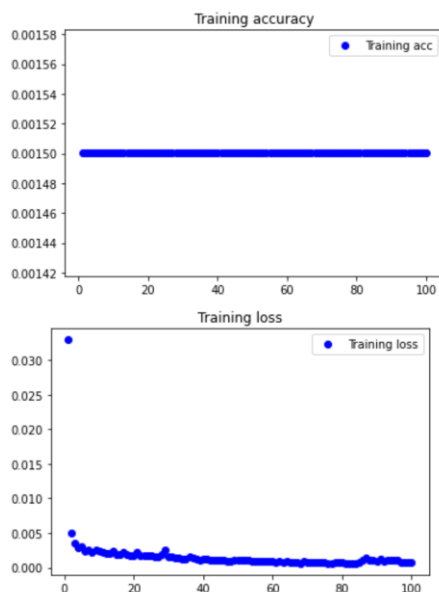
در مرحله بعد برای نرمالایز کردن داده ها از کتابخانه **sklearn** استفاده و **scale** را روی داده های آموزش و تست به صورت یکسان انجام دادم. در مرحله بعد **shape** داده های آموزش و تست را به ترتیب از **(1, 1393)** و **(1, 117)** به **(1, 1393)** و **(1, 117)** در آوردم تا بتوانیم راحت تر روی آن کار کنیم. سپس داده های تست و آموزش را برای دریافت داده های آینده تهیه کردم. برای این کار طبق داک متغیری تعریف و آنرا ۶۰ ست کردم تا داده های ۲ ماه قبل در داده های آینده تاثیر گذار باشند.

سپس مدل را طبق مدل خواسته شده تهیه و از لایه های **dropout** و **LSTM** و **flatten** و **dense** استفاده کردم. تعداد متغیر های مدل **71051** است.

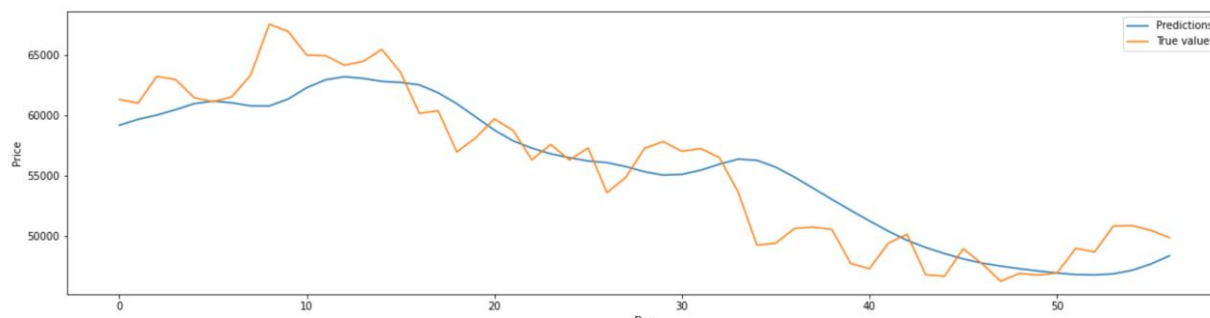
سپس در صد **epoch** و در **batch** های ۳۲ تایی مدل را ترین کردم. نتایج نهایی را مشاهده می کنید:

```
Epoch 96/100
42/42 [=====] - 5s 125ms/step - loss: 0.0011 - accuracy: 0.0015
Epoch 97/100
42/42 [=====] - 5s 124ms/step - loss: 8.1185e-04 - accuracy: 0.0015
Epoch 98/100
42/42 [=====] - 5s 125ms/step - loss: 7.7039e-04 - accuracy: 0.0015
Epoch 99/100
42/42 [=====] - 5s 126ms/step - loss: 7.9233e-04 - accuracy: 0.0015
Epoch 100/100
42/42 [=====] - 5s 125ms/step - loss: 7.8932e-04 - accuracy: 0.0015
```

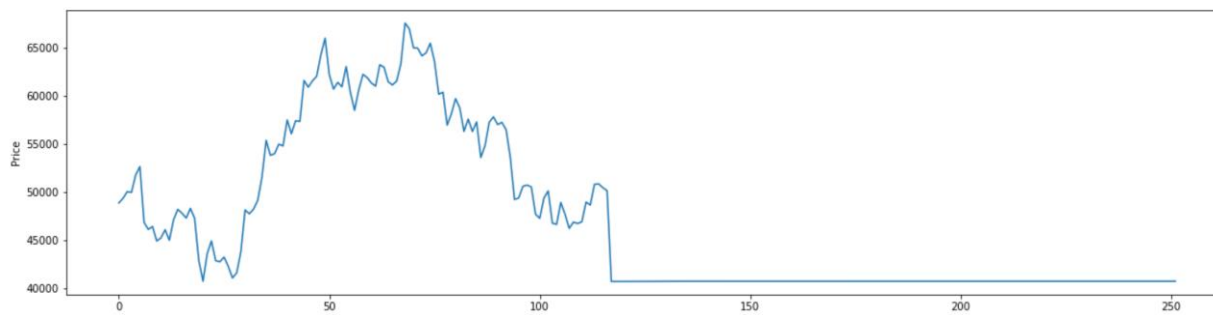
سپس نمودار های خطا و دقت مدل را رسم کردم. از آنجا که تعداد داده های مدل نسبت به عمق مدل کم است دقت خیلی کمی داریم و نیاز به داده های بیشتری داریم:



سپس مدل را روی داده های تست **evaluate** کردم و نتایج پیش بینی شده را با نتایج واقعی در کنار هم رسم کردم.



در مرحله آخر نیز از پیش بینی شبکه استفاده و پیش بینی مدل برای ۹۰ روز آینده را رسم کردم:



افزایش متغیر تعریف شده باعث افزایش تعداد داده های شبکه برای پیش بینی آینده میشود و در بیشتر مواقع نتیجه بهتر و دقیق تری برای ما به همراه دارد اما در اینجا هم زمان را بالا می برد هم ممکن است نتیجه عکس به همراه داشته باشد. چون هم ممکن است وزن داده ها برابر نباشد و در یک بازه اتفاق های مهم تری افتاده باشد یا از سوی دیگر قیمت دست خوش تغییر های بی ارزش شده باشد که برای پیش بینی آینده مناسب نباشد. از نظر این درس نیز **vanishing** ممکن است پیش بیاید و اثر روز های قبل خیلی کم شود و عملا بی تاثیر باشد.

اگر خیلی هم کم باشد فقط رفتار روز های اخیر محاسبه می شود که مناسب نیست و ممکن است این آثار را بیش از اندازه در آپدیت شبکه استفاده کند.

منابع: لینک و ویدیو های قسمت RNN درس

سوال دوم)

در این سوال هدف رمزگشایی متن رمزگذاری شده توسط شبکه RNN است. برای این هدف ابتدا متن های رمزگذاری و معادل آن ها را از فایل داده شده دریافت و جداسازی کردم. برای خواندن اطلاعات آن را در **drive** قرار داده و سپس آن را **mount** کردم و از آدرس آن در **drive** آن را دریافت کردم. سپس طبق نکته تذکر داده شده ابتدا طول آرگومان های دوم فایل را برابر با ۱۰ کردم و آنها که طول کمتر داشتند را با ' ' پر کردم.

سپس شبکه موردنظر را با لایه های GRU با ۱۲۸ نورون و **repeatvector** و **dense** تشکیل دادم.

```
layers=[tensorflow.keras.layers.GRU(units=128, input_shape=(10, 27)),
        tensorflow.keras.layers.RepeatVector(10),
        tensorflow.keras.layers.GRU(units=128, return_sequences=True),
        tensorflow.keras.layers.Dense(units=27, activation='softmax')]
```

در مرحله بعد از آنجا تعداد داده ها بالا و ران شدن شبکه RNN زمان بر است از سیو مدل های هر **epoch** استفاده کردم. ابتدا محل ذخیره آنرا در درایو ست کردم و سپس اگر در اولین ایپاک بودیم مدل را تشکیل دادم و ترین کردم نتیجه را ذخیره کردم و در ایپاک های بعدی از سیو مدل قبلی استفاده کردم. تمامی مدل های هر ایپاک نیز ذخیره شد. مدل را در ده **epoch** و با **batch** های ۳۲ تایی آموزش دادم. در مرحله آخر نیز مدل را **evaluate** کردم.

نتایج هر ایپاک شبکه را در زیر مشاهده می کنید. اعداد نشان از یادگیری خوب شبکه بر روی داده های آموزش است.

```
4759/4759 [=====] - 106s 21ms/step - loss: 0.0281 - accuracy: 0.3135
Epoch 1/10
4759/4759 [=====] - 106s 21ms/step - loss: 1.5484 - accuracy: 0.5333
Epoch 2/10
4759/4759 [=====] - 102s 22ms/step - loss: 0.6995 - accuracy: 0.7787
Epoch 3/10
4759/4759 [=====] - 102s 21ms/step - loss: 0.3895 - accuracy: 0.8803
Epoch 4/10
4759/4759 [=====] - 102s 21ms/step - loss: 0.2722 - accuracy: 0.9181
Epoch 5/10
4759/4759 [=====] - 102s 21ms/step - loss: 0.2213 - accuracy: 0.9336
Epoch 6/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.1949 - accuracy: 0.9417
Epoch 7/10
4759/4759 [=====] - 105s 22ms/step - loss: 0.1774 - accuracy: 0.9468
Epoch 8/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.1666 - accuracy: 0.9498
Epoch 9/10
4759/4759 [=====] - 105s 22ms/step - loss: 0.1573 - accuracy: 0.9527
Epoch 10/10
4759/4759 [=====] - 105s 22ms/step - loss: 0.1512 - accuracy: 0.9544
4759/4759 [=====] - 35s 7ms/step - loss: 0.1412 - accuracy: 0.9574
[0.1411917507648468, 0.9574047923088074]
```

سپس داده مورد نظر برای رمزنگاری را جداسازی و به شبکه دادم و خروجی آن را نمایش دادم:

```
z          wlove      qdeep      alearning
```

خروجی رمزگشایی شده کمی خطا داشت.

به همین دلیل سعی کردم شبکه دیگری هم تست کنم که فقط از لایه GRU استفاده شده باشد:

```
layers=[tensorflow.keras.layers.GRU(100, activation='relu', return_sequences=True,
                                     input_shape=(x_train.shape[1], x_train.shape[2])),
        tensorflow.keras.layers.GRU(75, activation='relu', return_sequences=True),
        tensorflow.keras.layers.GRU(27, activation='softmax', return_sequences=True)]
```

نتایج این مدل نیز در داده های آموزش بسیار خوب است که در زیر مشاهده می کنید:

```
4759/4759 [=====] - 105s 21ms/step - loss: 0.0115 - accuracy: 0.7513
Epoch 1/10
4759/4759 [=====] - 105s 21ms/step - loss: 0.2689 - accuracy: 0.9357
Epoch 2/10
4759/4759 [=====] - 101s 21ms/step - loss: 0.2368 - accuracy: 0.9445
Epoch 3/10
4759/4759 [=====] - 99s 21ms/step - loss: 0.2317 - accuracy: 0.9462
Epoch 4/10
4759/4759 [=====] - 100s 21ms/step - loss: 0.2298 - accuracy: 0.9467
Epoch 5/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.2279 - accuracy: 0.9470
Epoch 6/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.2262 - accuracy: 0.9475
Epoch 7/10
4759/4759 [=====] - 100s 21ms/step - loss: 0.2188 - accuracy: 0.9506
Epoch 8/10
4759/4759 [=====] - 108s 23ms/step - loss: 0.2124 - accuracy: 0.9529
Epoch 9/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.2117 - accuracy: 0.9530
Epoch 10/10
4759/4759 [=====] - 103s 22ms/step - loss: 0.2113 - accuracy: 0.9530
4759/4759 [=====] - 30s 6ms/step - loss: 0.2111 - accuracy: 0.9532
[0.2111031860113144, 0.9531919360160828]
```

خروجی این حالت نیز در زیر آورده شده است:

```
i          love      deep      learning
```

پس این نوع بهتر است و جواب کاملاً درستی را به ما ارائه میدهد.