**پروژه یادگیری ماشین(فاز سوم)**

سید محمد حاجی میرسعیدی

یاسمن ناظمی

بخش اول :

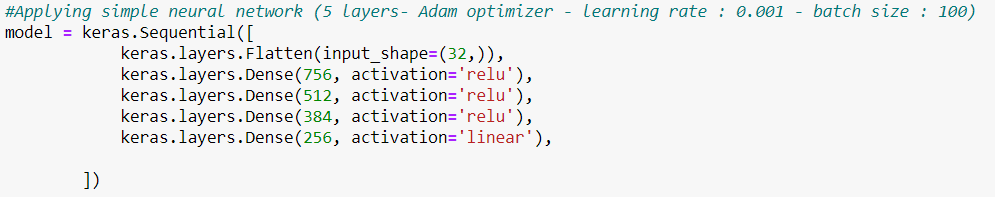
همانطور که در فاز دوم دیدیم متاسفانه نتایج خوبی برای این پروژه با روش های سنتی ماشین لرنینگ نگرفتیم برای همین با توجه به این موضوع که پیچیدگی مسئله زیاد بود انتظار میرفت که به خاطر اینکه شبکه های عصبی خاصیت غیر خطی بودن شدیدی دارند بتوان به وسیله آنها نتیجه بهتری گرفت ولی اینطور پیش نرفت و نتیجه خوبی حاصل نشد. تعداد داده هایی که ما در فاز اول تولید کردیم برابر ده هزار داده بود ولی به دلیل اینکه پیچیدگی مسئله زیاد است و ما باید پارامتر های مسئله را زیاد کنیم تعداد ده هزار داده تعداد خیلی کمی داده محسوب میشود . برای اینکه بتوانیم داده بیشتری داشته باشیم و مشکل اصلی را پیدا کنیم داده هایی به این صورت تولید کردیم که هر داده یک سیگنال سینوسی با یک فرکانس رندوم باشد و همینطور تبدیل فوریه آن یک تابع دلتا باشد. با این روش مسئله را بسیار ساده کردیم تا بنوان تعداد زیادی داده از آن تولید کنیم. تعداد داده های که با این روش تولید کردیم برابر یک میلیون داده بود ولی بازهم نتیجه تغییری نکرد و تابع هزینه پس از چند ایپاک دیگر تغییری نمیکند. هایپر پارامتر های مسئله را تغییر دادیم به این صورت که لرنینگ ریت را از 0.0001 تا 1 تغییر دادیم - تعداد لایه ها و نود های لایه ها و همچنین سایز batch -ولی باز همچنان مسئله تغییری نمیکرد .برای یک میلیون داده همزمان تابع هزینه داده های یادگیری و همچنین تست تغییری نمیکرد( تابع هزینه داده های تست زیاد نمیشد) اوور فیت رخ نداده بود برای همین نیازی به دراپ اوت یا رگولاریزیشن نبود ولی ما آنها را نیز تست کردیم ولی تغییری در نتیجه حاصل نشد:

ما 5 شکل شبکه های عصبی اعم از LSTM, Simple ANN ,CNN,GRU,RNN را امتحان کردیم ولی نتایج آنها تفاوت چندانی با یکدیگر ندارد. در حالتی که داده های جدید و در عین حال ساده تری را تولید کردیم ورودی و خروجی تفاوت جزیی با قبل پیدا کردند به این صورتکه ورودی همان 32 تایی به معنی اینکه 32 فیچر داشت و خروجی 128 تایی شد که با نتیجه ی سری قبلی که 256 تایی است متفاوت است.

نتایح برای هر دو سری داده های تولید شده در صفحه ی گیت هاب موجود است. عنوان داده های جدید ی که تولید کردیم با نام Generating data در صفحه ی گیت هاب وجود دارد. این داده ها فقط با شبکه ساده ANN ارزیابی شده اند.

DNN:

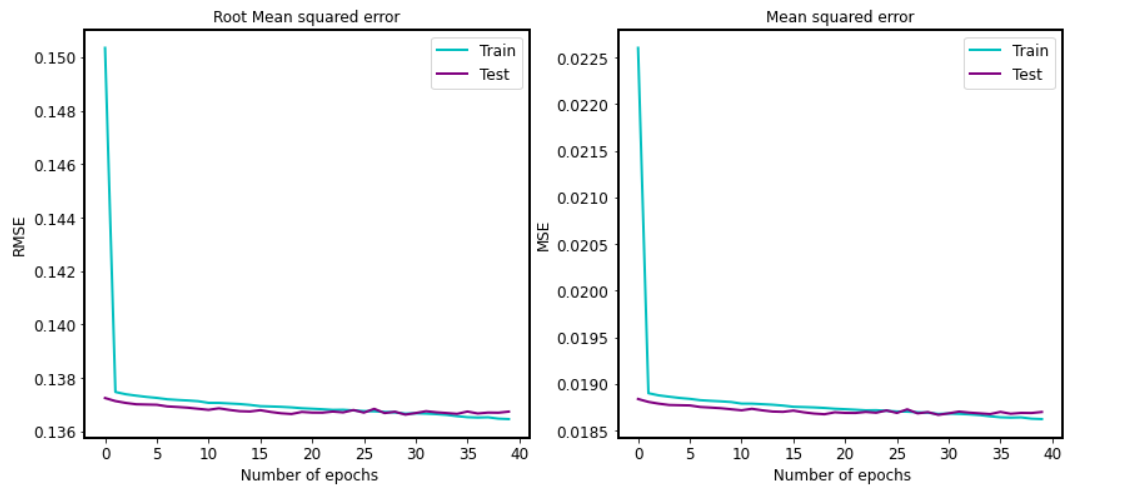
در این روش ابتدا با 4 لایه مسئله را بررسی کردیم که چون مسئله پیچیده بود تعداد لایه ها را یکی زیاد کردیم و بعد از آن تعداد نود ها را نیز به صورت زیر درست کردیم:



در این قسمت لرنینگ ریت های متفاوت از 0.0001 تا 1 با بچ سایز های متفاوت که در کد موجود هستند و همینطور از اپتیمایزر های متفاوت شامل Adamو SGDوRmsprop را هم تست کردیم ولی نتایج تفاوت چشمگیری نداشتند.

نتایج یکی از مدل ها

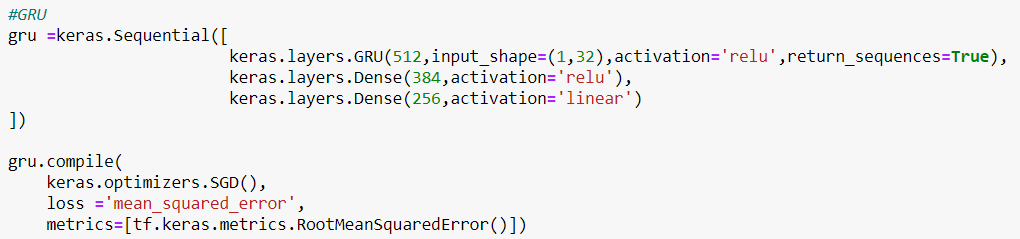
layers- Adam optimizer - learning rate : 0.00012 - batch size : 1005

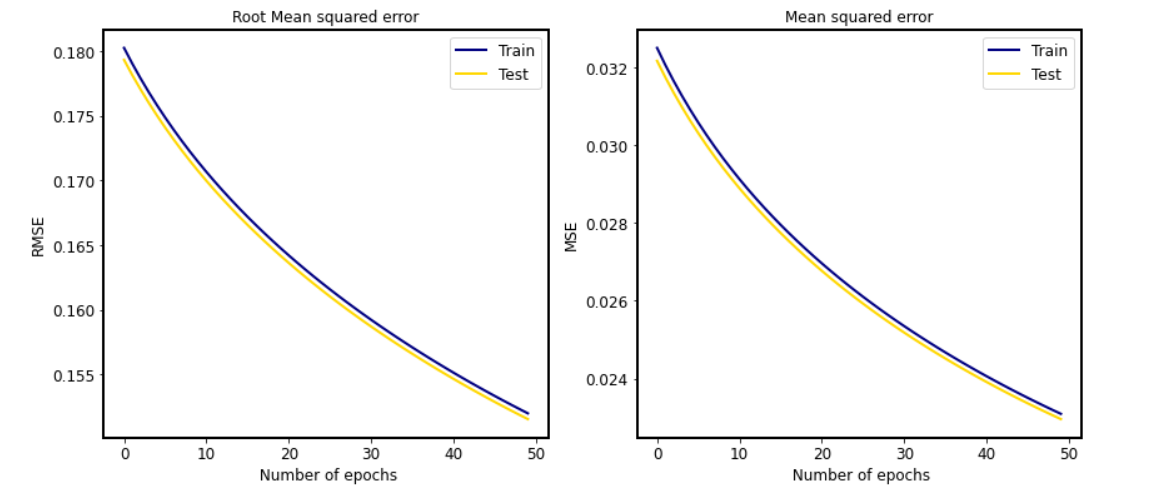


GRU:

مدلی که طراحی کردیم به صورت زیر است:

SGD optimizer - learning rate : 0.01 - batch size : 100

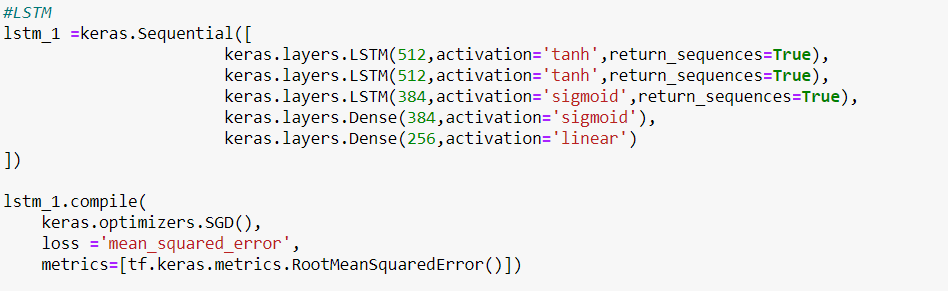


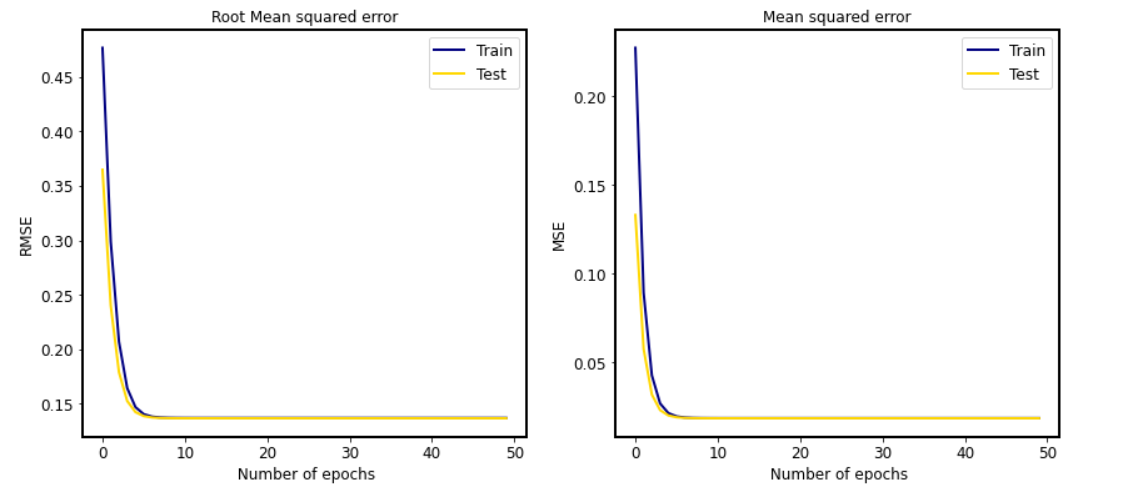


LSTM:

برای این مدل داریم:

SGD optimizer - learning rate : 0.01 - batch size : 100

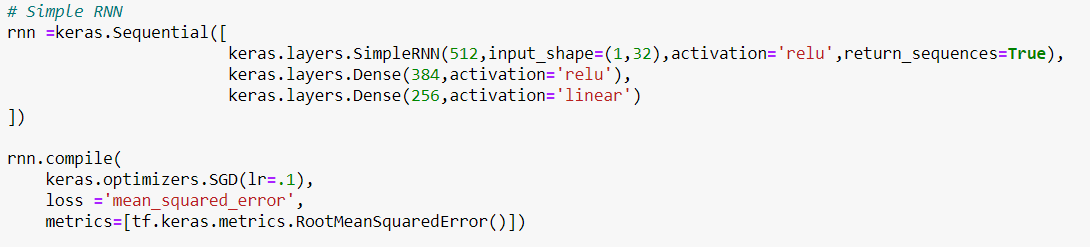


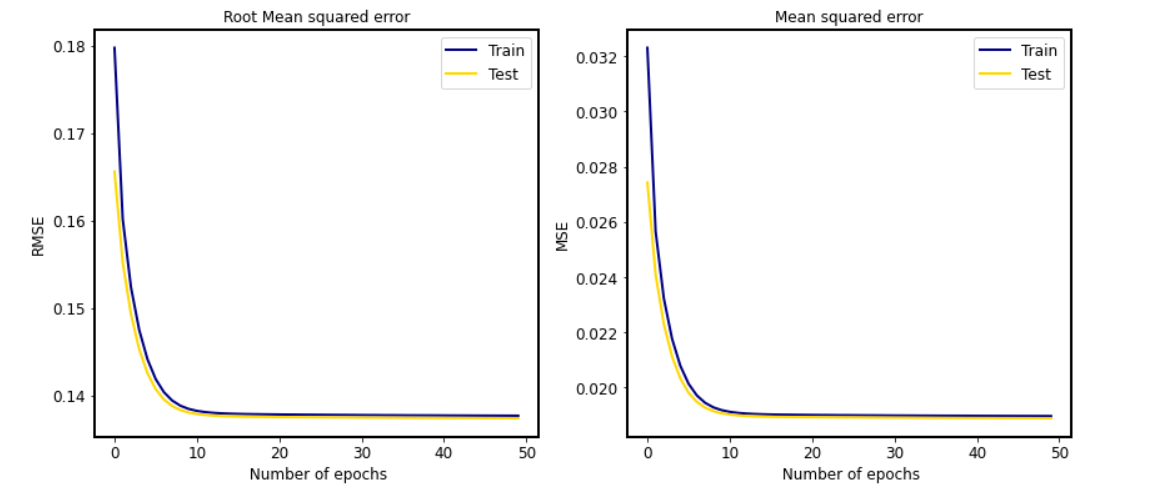


RNN:

برای این مدل داریم:

SGD optimizer - learning rate : 0.1 - batch size : 100





CNN:

مدل آن به صورت زیر است:

Adam optimizer - learning rate : 0.01 - batch size : 100

