فهرست گزارش سوالات

سوال 1 – (Regression)MLP

الف) در این قسمت ستون هایی که بیش از ۳۰ درصد داده NA دارند را به عنوان ویژگی های کم اهمیت حذف کرده ایم.در این صورت ابعاد داده ما ۷۶ ×۱۴۶۰ می شود.حال ستون های Categorical را در نظر گرفته و آن ها را به داده عددی کد می کنیم و برای آن ها NA به منفی یک مپ می گردد. در ادامه و برای ستون های نومریکال نیز به جای Missing values از میانگین آن ستون جایگذاری می کنیم.در نهایت و پس از تقسیم بندی داده به تست و ترین آن ها را توسط MinMaxscalar نرمالایز می کنیم. حالا داده ها آماده می باشند.

ب)ساختار های در نظر گرفته برای اینش بکه در شکل های ۱ و۲ آورده شده. در این قسمت از در لایه های میانی از تابع فعالساز Relu استفاده شده و برای لایه آخر یک بار از Linear و بار دیگر از tanh استفاده شد. همچنین بعد از هر لایه یک لایه droupout به منظور عدم اوورفیت کردن مدل در نظر گرفته شده است.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 75)                5700
_________________________________________________________________
dropout_12 (Dropout)         (None, 75)                0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               38912
_________________________________________________________________
dropout_13 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 513
=================================================================
Total params: 45,125
Trainable params: 45,125
Non-trainable params: 0
```

شکل۱ -۱) اولین ساختار در نظر گرفته شده

Layer (type)	Output Shape	Param #
dense_21 (Dense)	(None, 75)	5700
dropout_14 (Dropout)	(None, 75)	0
dense_22 (Dense)	(None, 512)	38912
dropout_15 (Dropout)	(None, 512)	0
dense_23 (Dense)	(None, 256)	131328
dense_24 (Dense)	(None, 1)	257

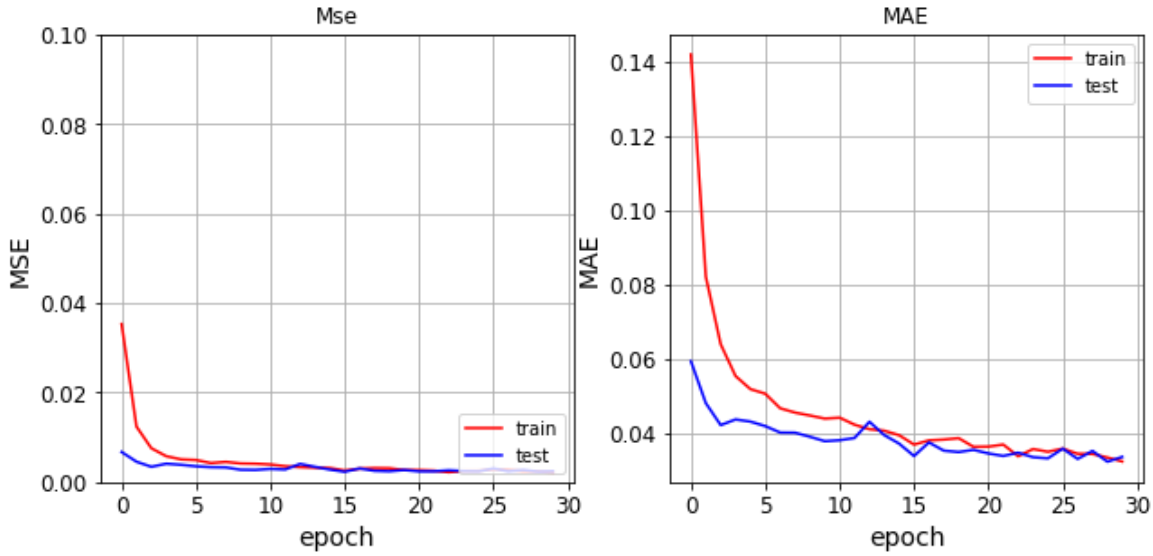
Total params: 176,197
Trainable params: 176,197
Non-trainable params: 0

شکل ۱ – ۲) دومین ساختار در نظر گرفته شده

در این قسمت در اولین ساختار یک شبکه با دو لایه مخفی داریم که با توجه به عملکرد مناسب آن با در نظر گرفتن پارامترهای خطا از آن استفاده می شود.هم چنین در لایه ی آخر از دو تابع فعالساز tanh و Linearاستفاده شد که تابع Linearدر واقع مقدار دریافتی در ورودی خود را بدون تغییر در خروجی قرار می دهد.پس مدل نهایی انتخاب شده دارای دو لایه ی مخفی با تابع فعالساز Relu و لایه ی نهایی تابع فعالساز Linear دارد.

ج) با توجه به در نظر گرفتن مدل نهایی فوق نتایج زیر بدست می آید.

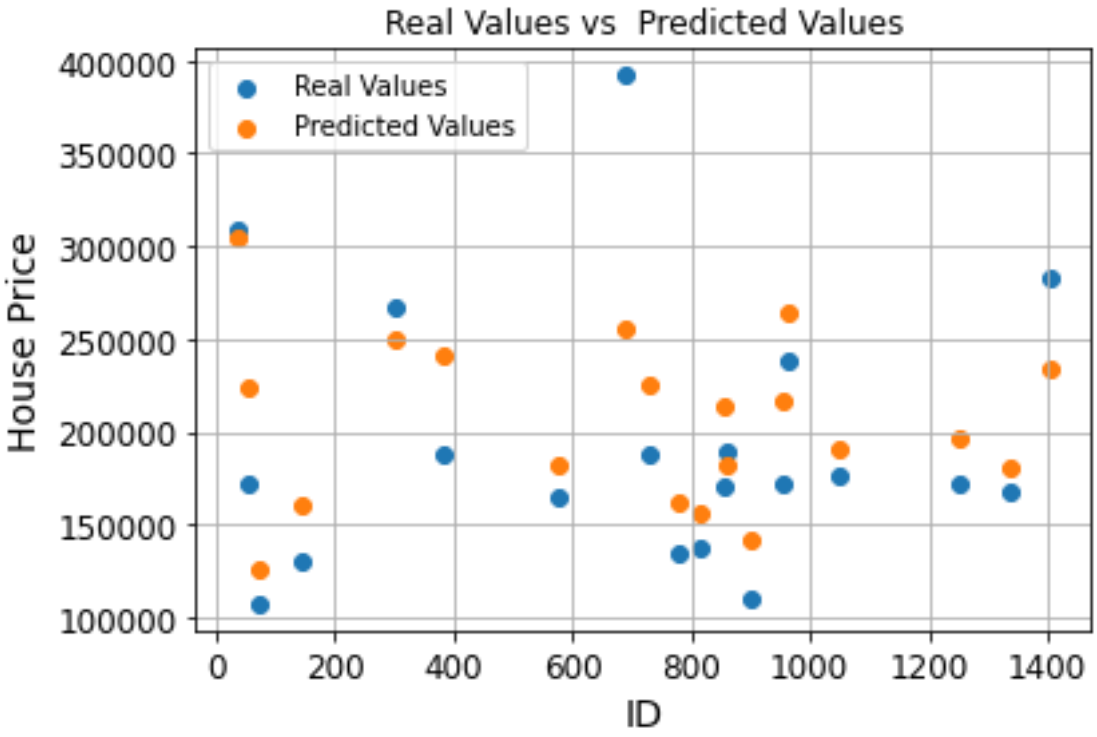
شبکه را در ۳۰ ایپاک آموزش می دهیم، در این قسمت MSE بعنوان Loss در نظر گرفته شده است. نتایج زیر بدست می آیند :



شکل ۱ـ۳) نتایج با در نظر گرفتن Loss=MSE

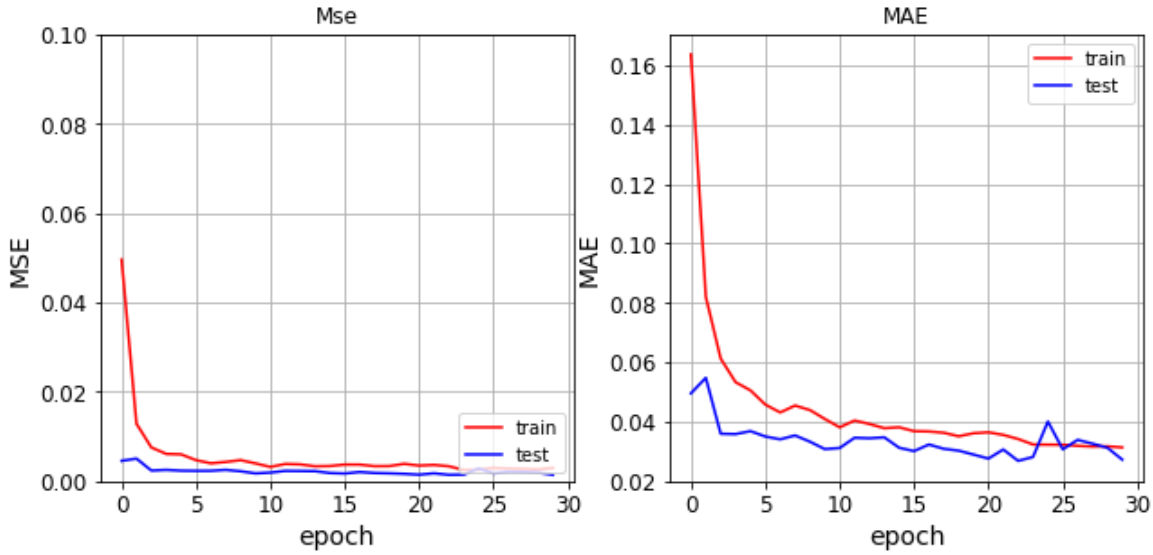
بنظر می رسد که در حدود ۱۵ ایپاک به مقدار بهینه ی خود دست پیدا می کنیم.

حال برای داده ی تست به ازای برخی داده بصورت رندوم (۲۰ تا) نمودار مقادیر واقعی و تخمین زده شده بر حسب Id رسم شده است.



شکل ۱ـ۴)مقادیر واقعی و تخمین زده شده بر حسب ID

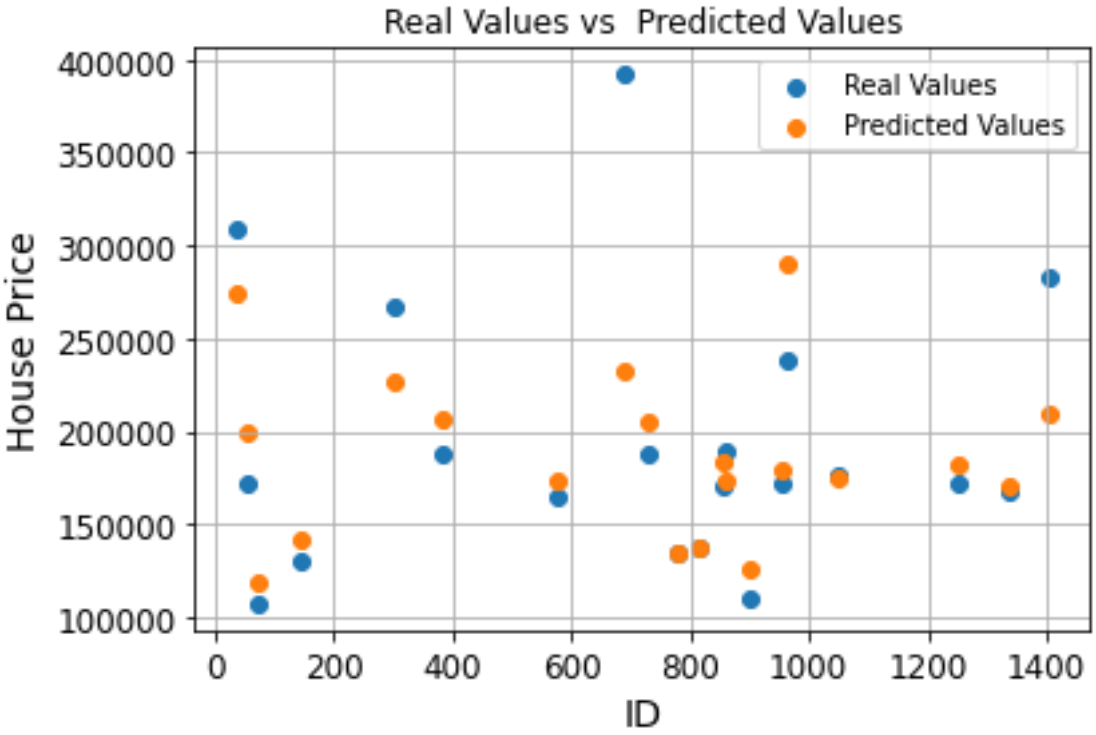
د) شبکه را در ۳۰ ایپاک آموزش می دهیم، در این قسمت MAE بعنوان Loss در نظر گرفته شده است. نتایج زیر بدست می آیند:



شکل ۱ـ۵) نتایج با در نظر گرفتن Loss=MAE

بنظر می رسد که در حدود 20 ایپاک به مقدار بهینه ی خود دست پیدا می کنیم.

حال برای داده ی تست به ازای برخی داده بصورت رندوم (۲۰ تا) نمودار مقادیر واقعی و تخمین زده شده بر حسب Id رسم شده است.



شکل ۱ـ۶)مقادیر واقعی و تخمین زده شده بر حسب ID

ه) معیارهای MAE , MSEو تعریف های مربوط به آن در ادامه آورده شده است.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

در این بخش و بررسی نمودار های مربوط به دو قسمت قبل مشاهده می شود که با توجه به آنکه هر دو معیار به نوعی میانگین اندازه خطا (درMSE توان دو آن را) بررسی می کنند عملکرد تقریبا مشابه ای دارند.گرچه بنظر می رسد در نظر گرفتن MAE به عنوان Lossدر این قسمت می تواند مناسب تر باشد(الگوریتم بهتر پارامتر های شبکه را ترین می کند.)

سوال ۲ – (Classification) MLP

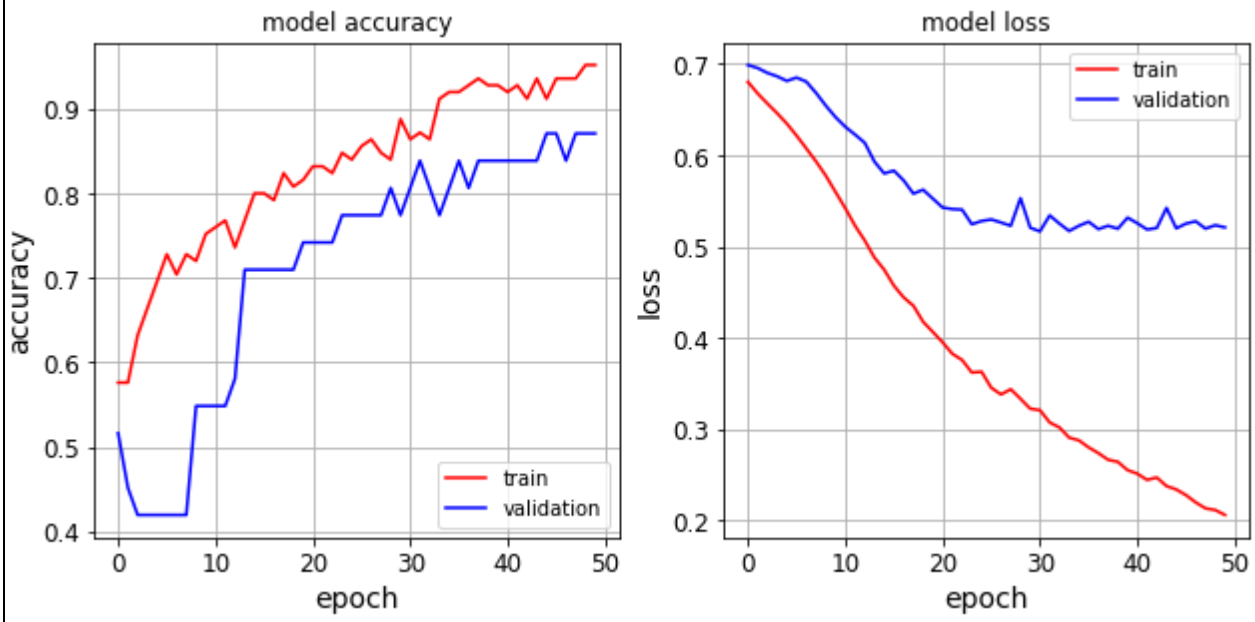
الف) ویژگی های X را نرمالایز کرده و Yرا به عنوان Label : (R↔1) و (M ↔0) در نظر می گیریم. برای تقسیم بندی داده، ابتدا داده را بطور رندوم ۲۵ درصد به تست اختصاص می دهیم. در ادامه با توجه به کم بودن تعداد داده ها از Cross Validation استفاده می کنیم (5 fold Cross Validation).
ساختار زیر را برای شبکه با دو لایه ی مخفی در نظر میگیریم :

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_30 (Dense)             (None, 20)                1220
_________________________________________________________________
dense_31 (Dense)             (None, 256)               5376
_________________________________________________________________
dense_32 (Dense)             (None, 1)                 257
=================================================================
Total params: 6,853
Trainable params: 6,853
Non-trainable params: 0
_________________________________________________________________
```

شکل ۲ـ۱) ساختار معماری شبکه

در لایه های میانی شبکه از تابع فعالساز relu استفاده شده ، و در لایه ی نهایی از sigmoid استفاده شد.

ب) معیار دقت را Accuracy در نظر گرفته و loss='binary_crossentropy' در نظر می گیریم.



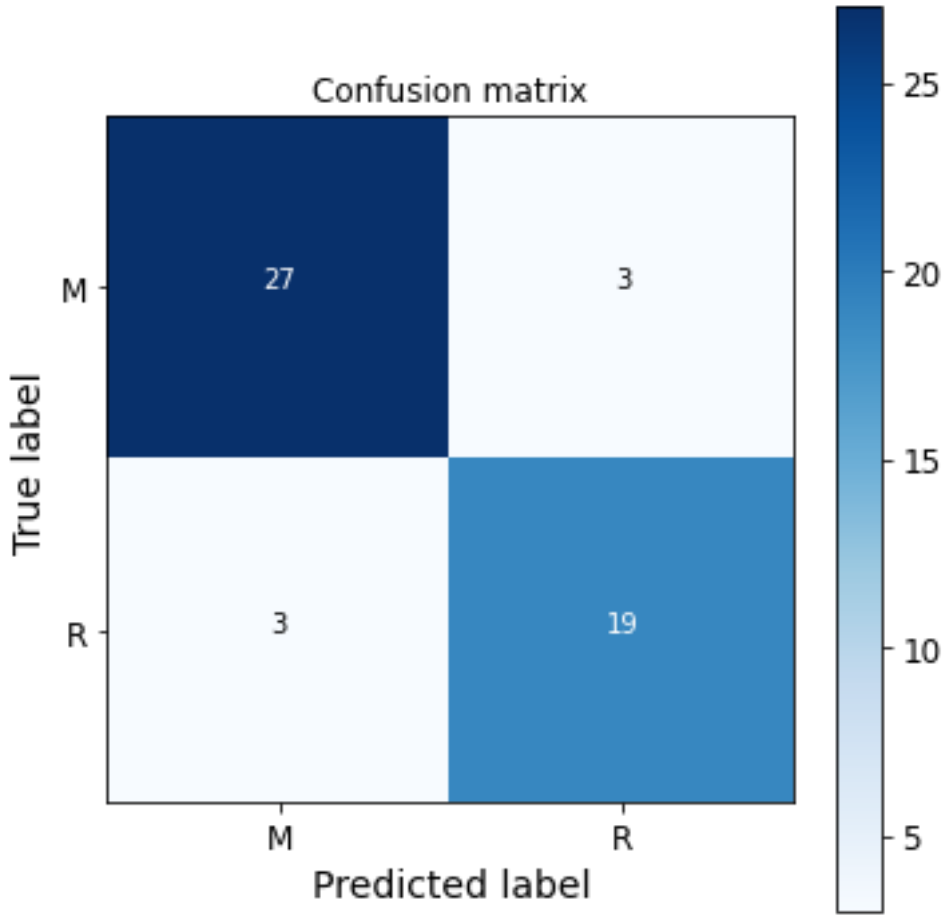
شکل ۲ـ۲) دقت و loss به ازای هر ایپاک

با توجه به نمودارهای بدست آمده در ۳۰ ایپاک به مقدار بهینه دست پیدا می کنیم.

ج) برای داده ی تست نتایج زیر بدست می آید:

Accuracy=88.46 %

Loss=0.31



شکل ۲-۳) Confusion Matrix برای داده تست

د) برای کلاسه بندی دو کلاسه از binary_crossentropy استفاده شده است. رابطه ی آن را به فرم

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}) + (1-y_i)\log(1-\hat{y}_i)]$$

در نظر میگیریم. هم چنین در صورت one hot بودن فضای خروجی sparse_binary_crossentropy استفاده می شود.می توان از Hinge Loss برای فضای خروجی (۱-و۱) می توان استفاده نمود.بهینه ترین حالت همان binary_crossentropy می باشد.

ه) معیار دقت استفاده شده Accuracy بود که مجموع لیبل های درست طبقه بندی شده به کل داده ها را نشان می دهد.اما به معیارهای دیگری نیز نیاز داریم تا در شرایطی بتوان عملکرد هر کلاس را نیز بررسی نمود.معیار های Precision و Recall را در اینجا در نظر میگیریم.

TP=true Positive , FN=False Negative

Recall=TP/(TP+FP)

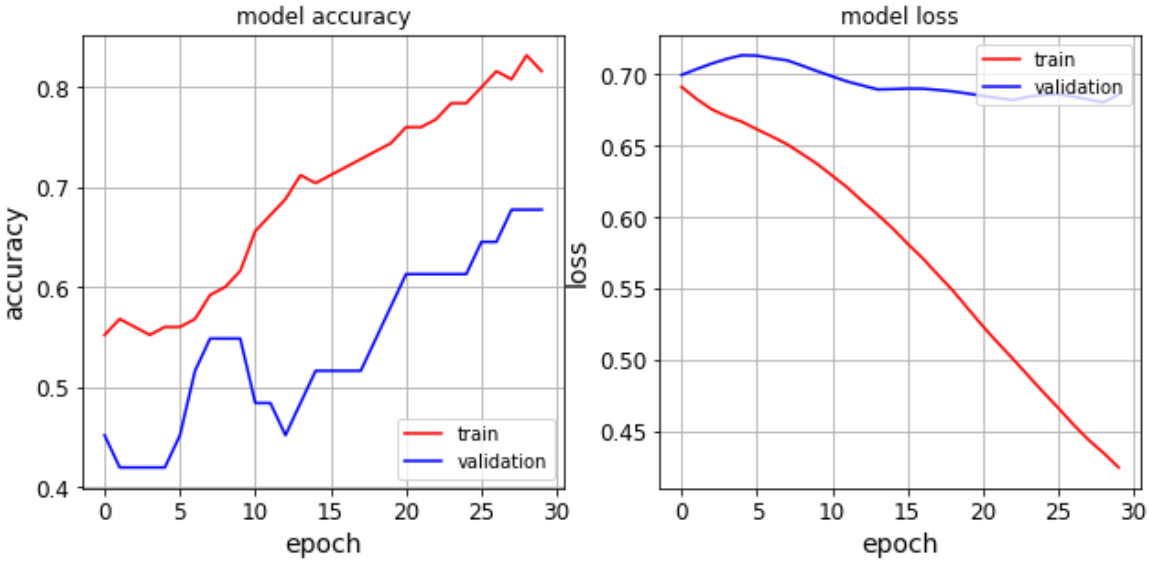
Precision=TP/(TP+FN)

که در این مثال مقادیر زیر بدست آمدند.

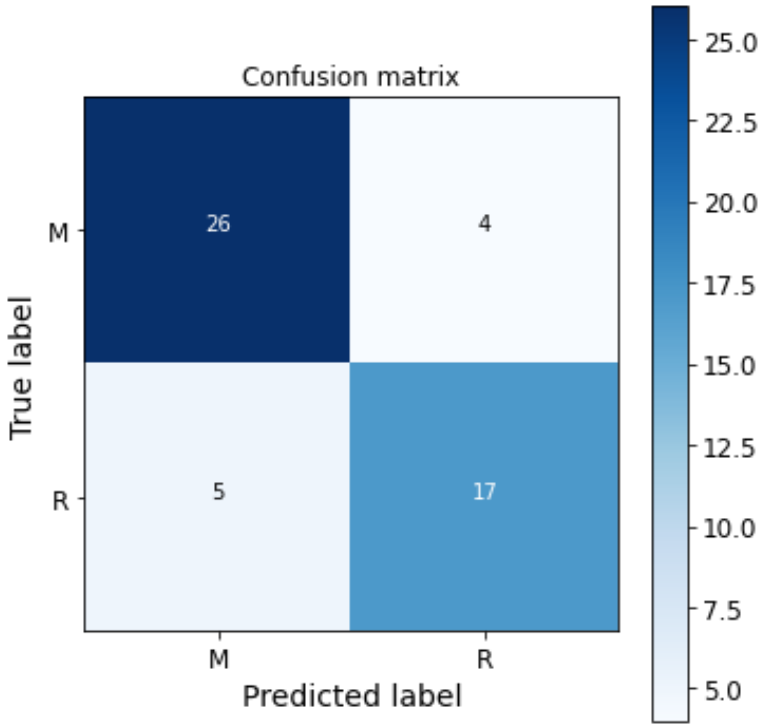
	precision	recall	f1-score	support
0.0	0.90	0.90	0.90	30
1.0	0.86	0.86	0.86	22

و) در این قسمت در model.fit یک پارامتر batch_size را مطابق اعداد ۳۲ و۶۴ و ۱۲۸ در نظر می گیریم، این اعداد بیان می کنند که در هر iteration چه قسمتی از داده ترین مورد استفاده قرار می گیرد. (به عنوان یک iterationدر داخل epoch)

1) batch_size=32

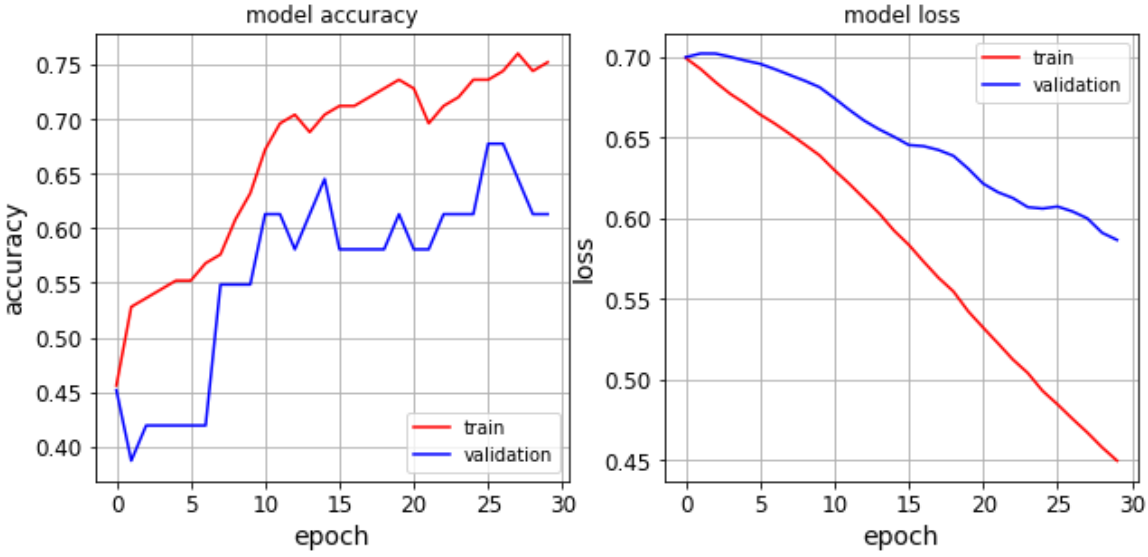


شکل ۲ـ۴) دقت و loss به ازای هر ایپاک batch size=32

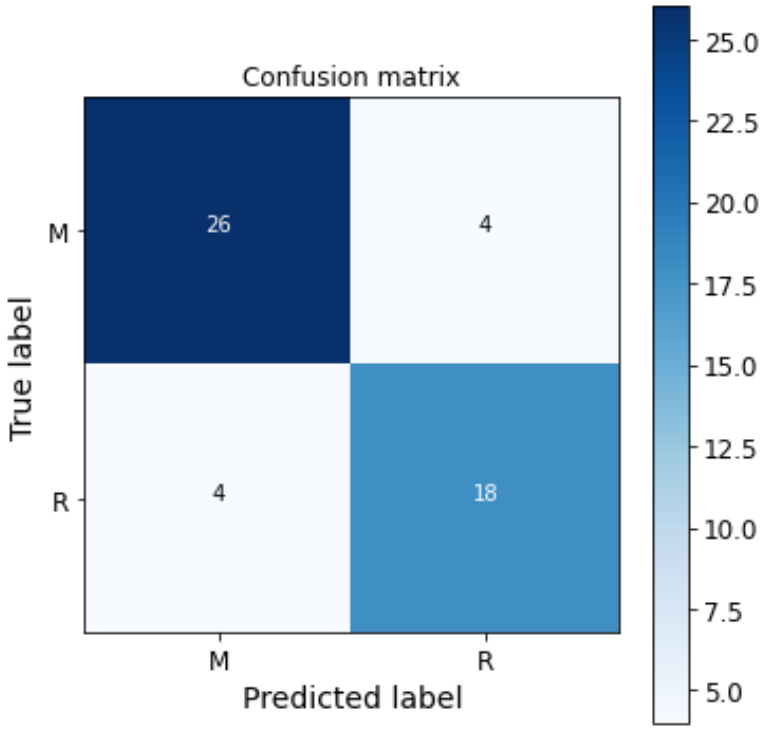


شکل ۲ـ۵) Confusion Matrix برای داده تست

2) batch_size=64

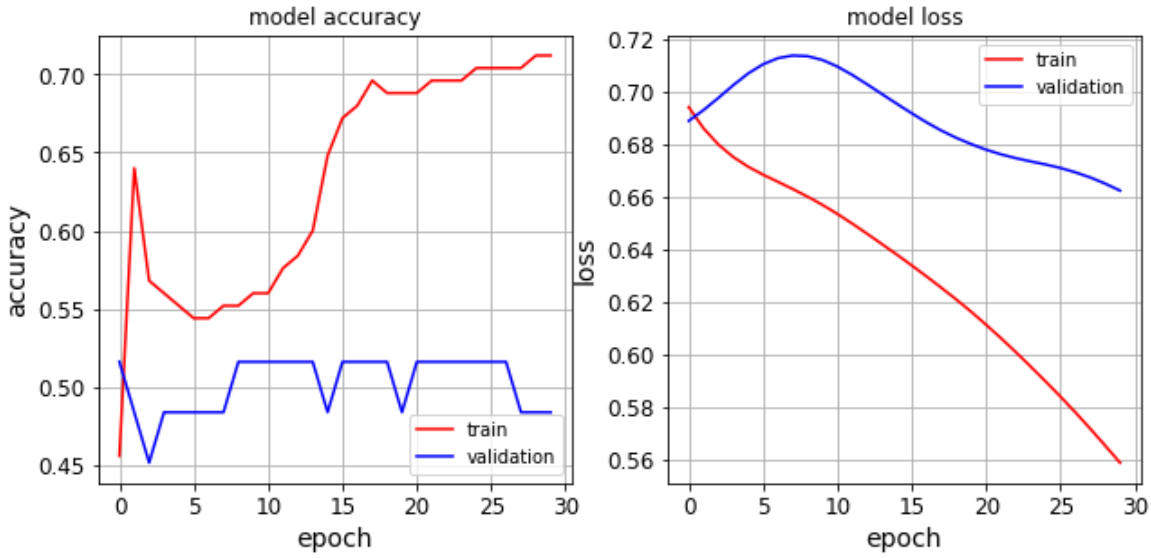


شکل ۲ـ۶) دقت و loss به ازای هر ایپاک batch size=64

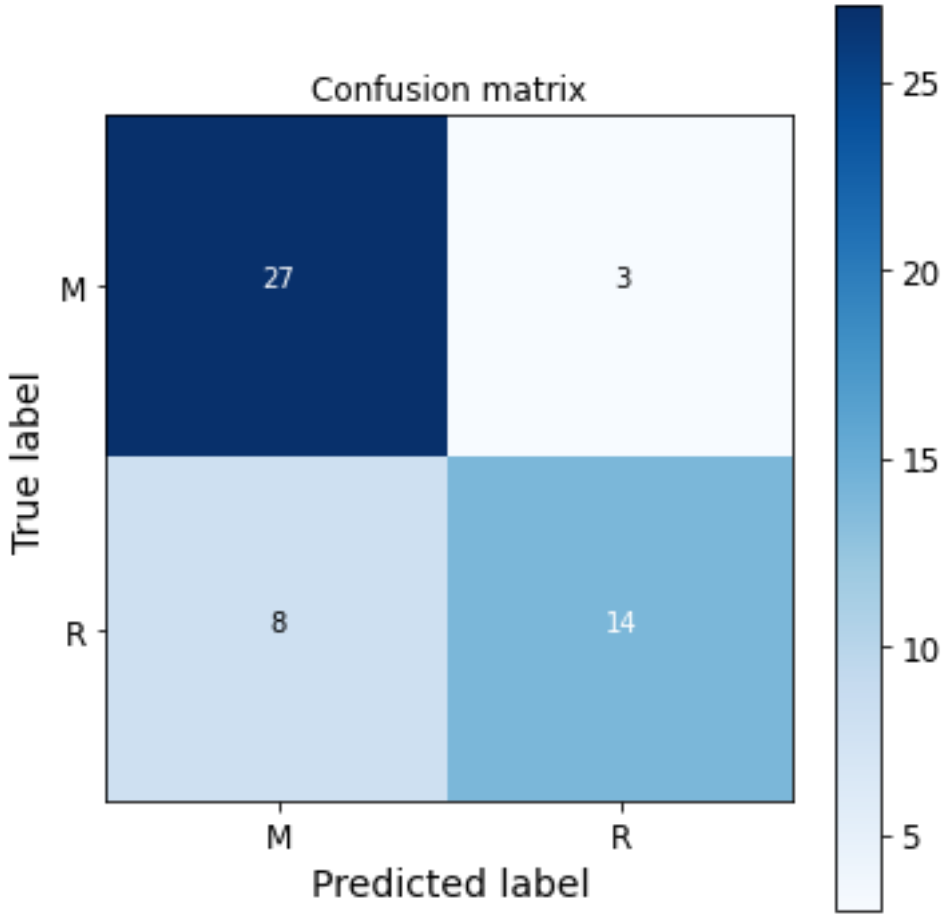


شکل ۲ـ۷) Confusion Matrix برای داده تست

3) batch_size=128



شکل ۲ـ۸) دقت و loss به ازای هر ایپاک batch size=128



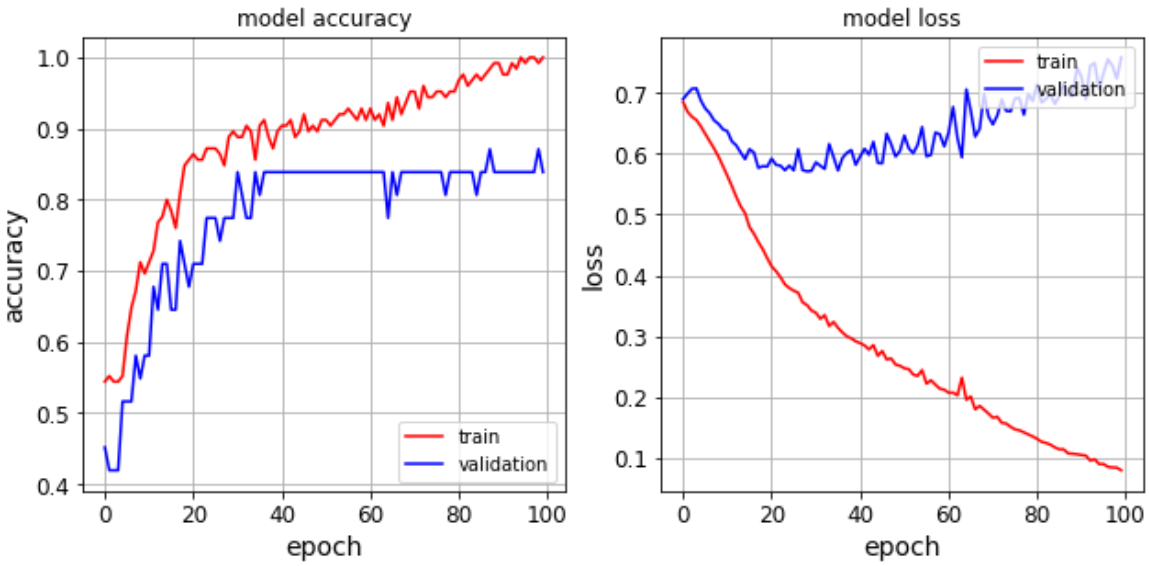
شکل ۲ـ۹) Confusion Matrix برای داده تست

بررسی نتایج فوق نشان می دهد با توجه به کم بودن تعداد داده انتخاب patch_size=128 نمی تواند انتخاب مناسبی برای این شبکه باشد و در ۳۰ ایپاک نتوانسته به خوبی مدل را آموزش دهد در رابطه با دو مدل دیگر در batch size=32 بنظر می رسد که تابع Lossبهتر کاهش پیدا می کند گرچه در ۶۴ نیز عملکرد مناسبی را برای این مدل داریم (از نظر دقت و خطا).

ح) epoch را معادل تعداد کل بار های اعمال تغیرات بر روی کل داده ی ترین در نظر می گیریم در حالیکه هر Iteration اعمال قوانین آپدیت بر روی یک ست پچ در نظر گرفته شده در هر مرحله می باشد.برای مثال در حالتی که batch_size=32 می باشد با توجه به آنکه تعداد سطرهای داده ترین ۱۲۵ می باشد ۴ ایتریشین در این مثال خواهیم داشت. به همین نحو برای حالت ۶۴ تعداد ایتریشن در هر ایپاک ۲ و در حالت ۱۲۸ این مقدار برابر یک ایتریشن می شود.

مقدار بهینه ی ایپاک را طوری در نظر می گیریم که خطا برای داده ترین و ولیدیشن کاهش یابد تا جایی که مقدار لاس افزایش یابد دیگر ادامه نمی دهیم.

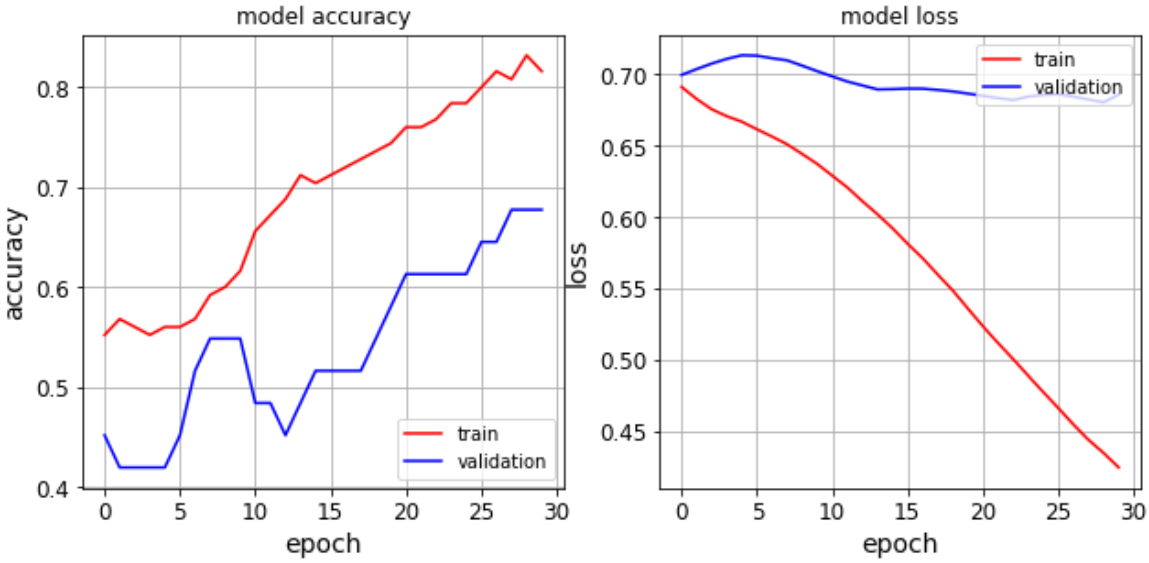
با افزایش تعداد ایپاک ها مدل برای داده ترین می تواند بطور کامل ترین شود و خطا و دقت به مقدار کامل و بهینه ی خود برسند، در حالیکه ولیدیشن نشان می دهد که نمی تواند ترین را دنبال کند و خطای آن افزایش می یابد گرچه نتایج برای داده تست بنسبت مناسب می باشد که می تواند ناشی از مدل داده ها یا جداسازی نامناسب تست و ترین باشد. این نشان از اوورفیت کردن مدل ما بر روی داده ی ترین (آموزشی) می باشد. مثال زیر خطا و دقت را برای batch_size=32 و در ۱۰۰ ایپاک نمایش می دهد.



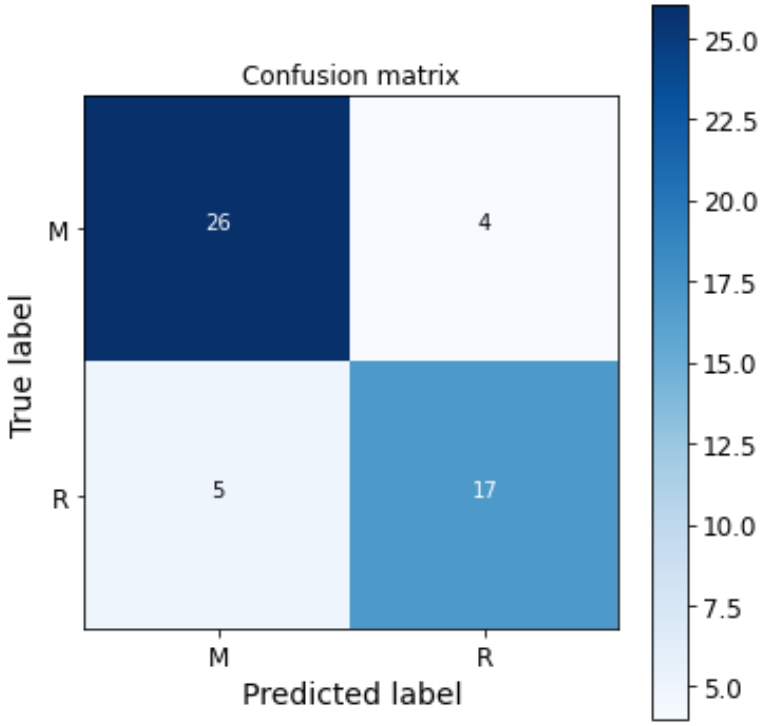
شکل ۲ـ۱۰) دقت و loss به ازای هر ایپاک batch size=32

ط)نتایج قبلی در لایه های میانی Relu بود.حال در حالت Batch_size=32 در لایه های میانی یکبار tanh و بار دیگر sigmoid گذاشته و نتایج را بیان می کنیم.

I) Activation_func = Linear

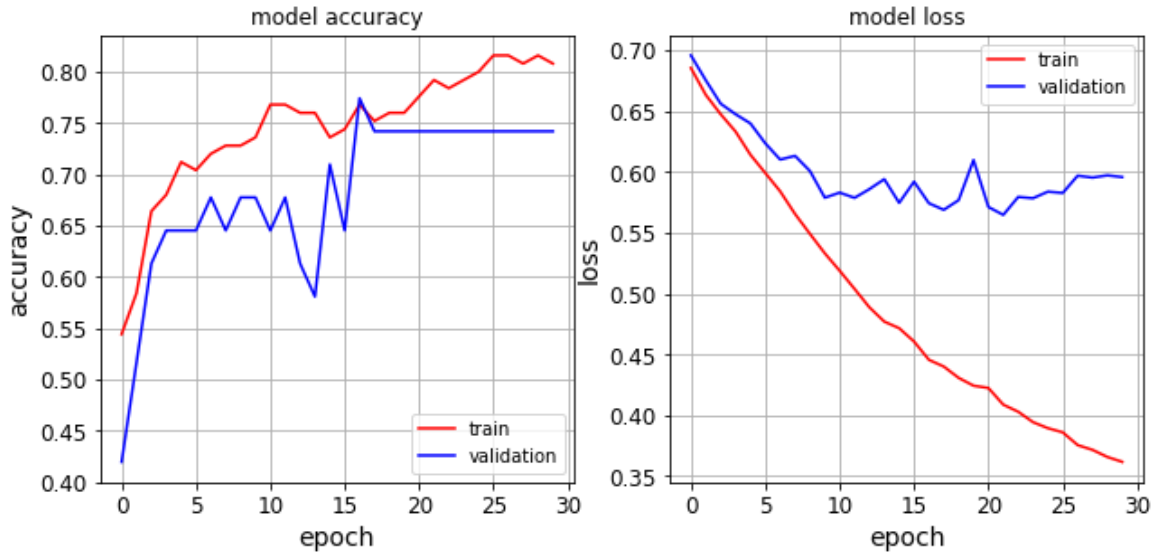


شکل ۲ـ۱۱) دقت و loss به ازای هر ایپاک batch size=32

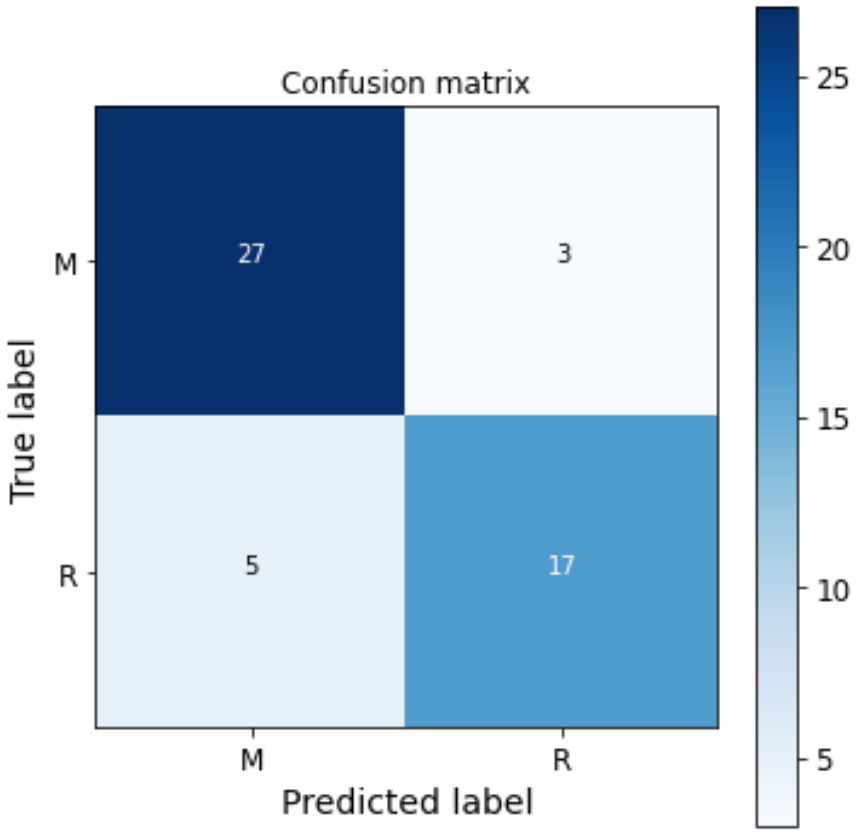


شکل ۲ـ۱۲) Confusion Matrix برای داده تست

II) Activation_func = tanh

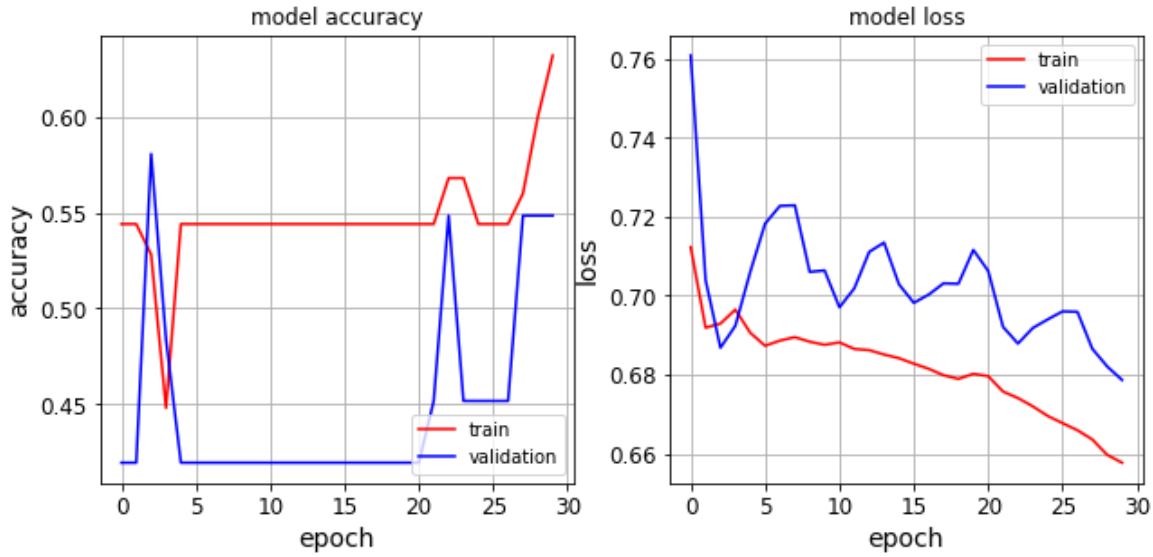


شکل ۲ـ۱۳) دقت و loss به ازای هر ایپاک batch size=32و تابع فعالساز tanh

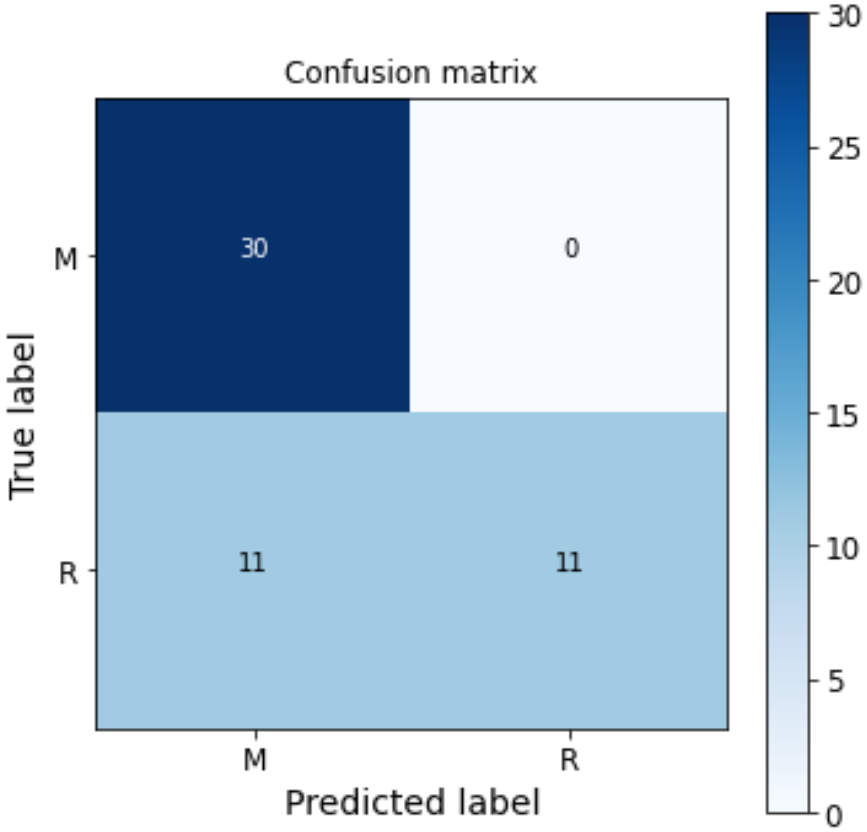


شکل ۲ـ۱۴) Confusion Matrix برای داده تست

III) Activation_func = Sigmoid



شکل ۲ـ۱۵) دقت و loss به ازای هر ایپاک batch size=32و تابع فعالساز Sigmoid



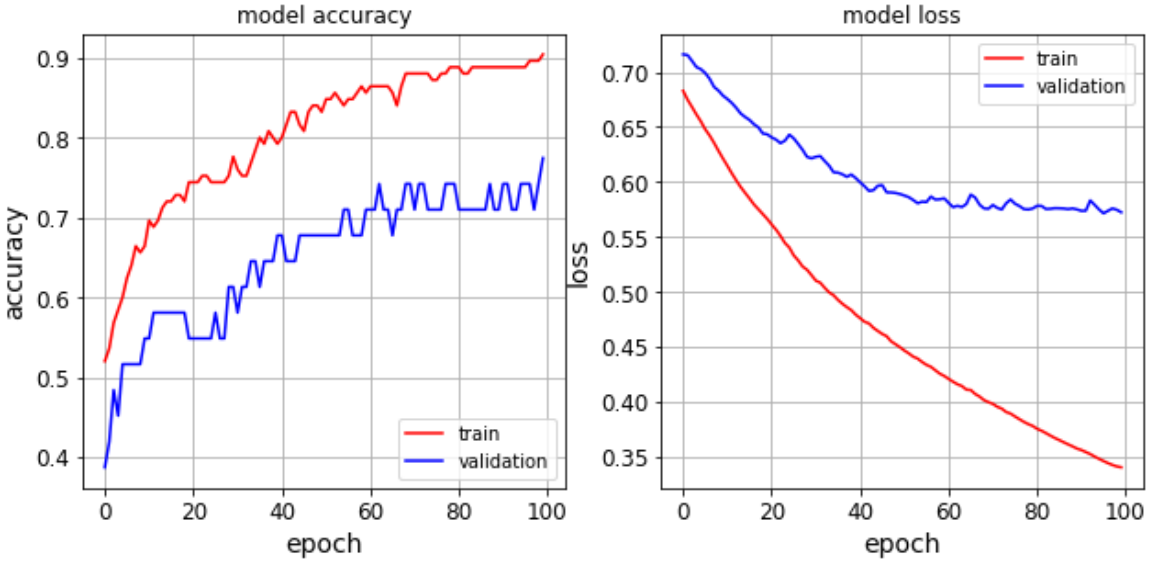
شکل ۲ـ۱۶) Confusion Matrix برای داده تست

با بررسی نتایج فوق مشاهده می شود خروجی های مرتبط با Relu ,Tanh دقت و خطا تقریبا مشابه هم می باشند در حالیکه با قرار دادن Sigmoidبا خطای زیادی مواجه شدیم.تابع Relu به ازای ورودی های مثبت عینا عبورمی دهد و منفی ها را صفر میکند در این تابع و حسنی که نسبت به حالت tanh داریم عدم اشباه در گرادیان مربوط آن می باشد که به Convergenceکمک می کند در حالیکه تابع tanh smooth می باشد و مشتق های مرتبه بالای آن نیز برای تحلیل های ریاضی موجود می باشد.در این مصال هر دوی آن ها عملکرد مناسبی دارند.اما در مجموع sigmoid , tanh می توانند مشکلاتی را در Gradient بوجود آورند.

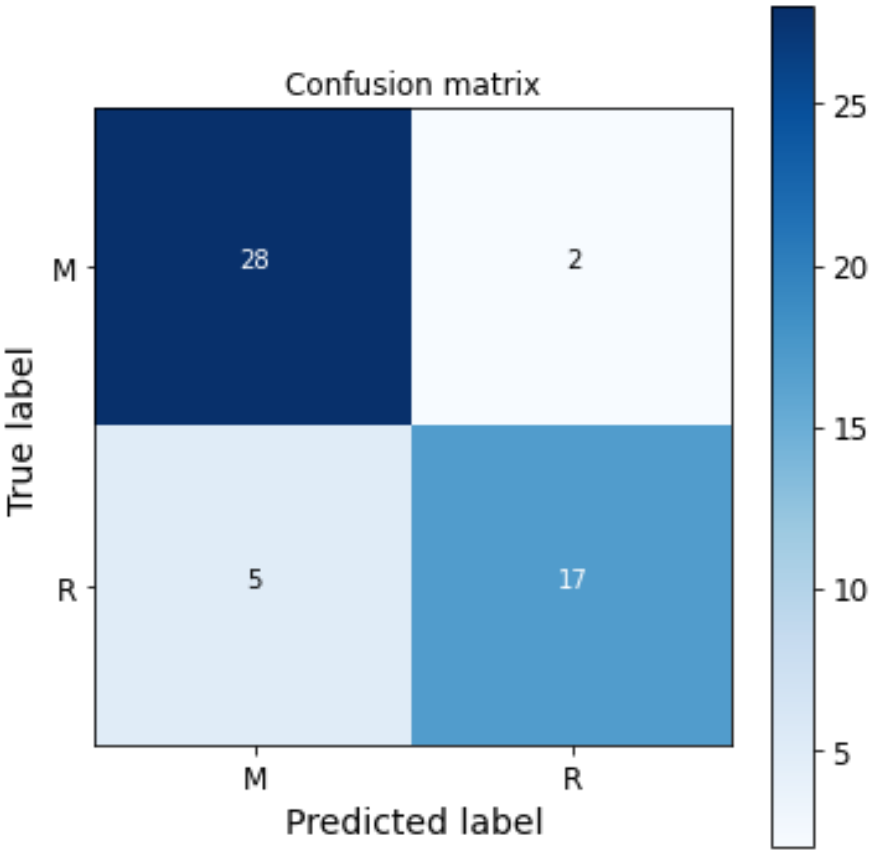
ی)تمام تحلیل های فوق دارای دو لایه ی مخفی بودند حال در این قسمت قصد داریم در حالت ۳ و ۱ لایه ی مخفی نیز شرایط را بررسی نماییم

۱) شبکه با یک لایه مخفی:

در این قسمت با در نظر گرفتن تنها یک لایه قبل از نورونت کلاسیفایر با ۲۰ نورون به نتیجه ی زیر برای داده ی تست دست پیدا می کنیم.در اینجا با شیب ملایم تری به سمت فیت شدن و در تعداد بالاتری ایپاک به این مهم دست پیدا می کنیم.(شکل زیر نتایج مرتبط با دقت و خطا را در ۱۰۰ ایپاک نمایش می دهد.)



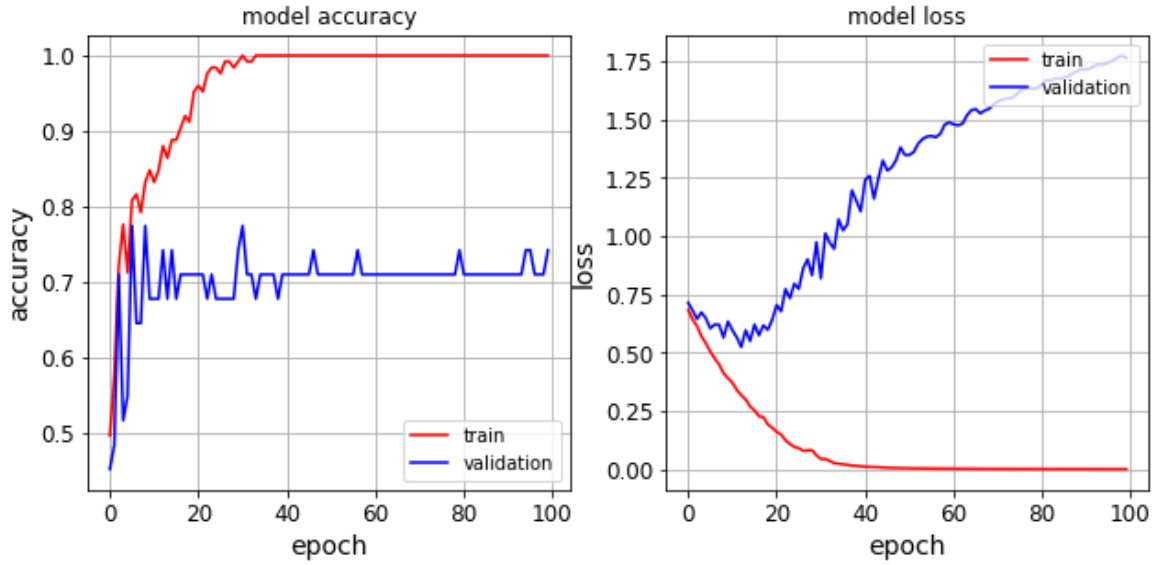
شکل ۲ـ۱۷) دقت و loss به ازای هر ایپاک batch size=32



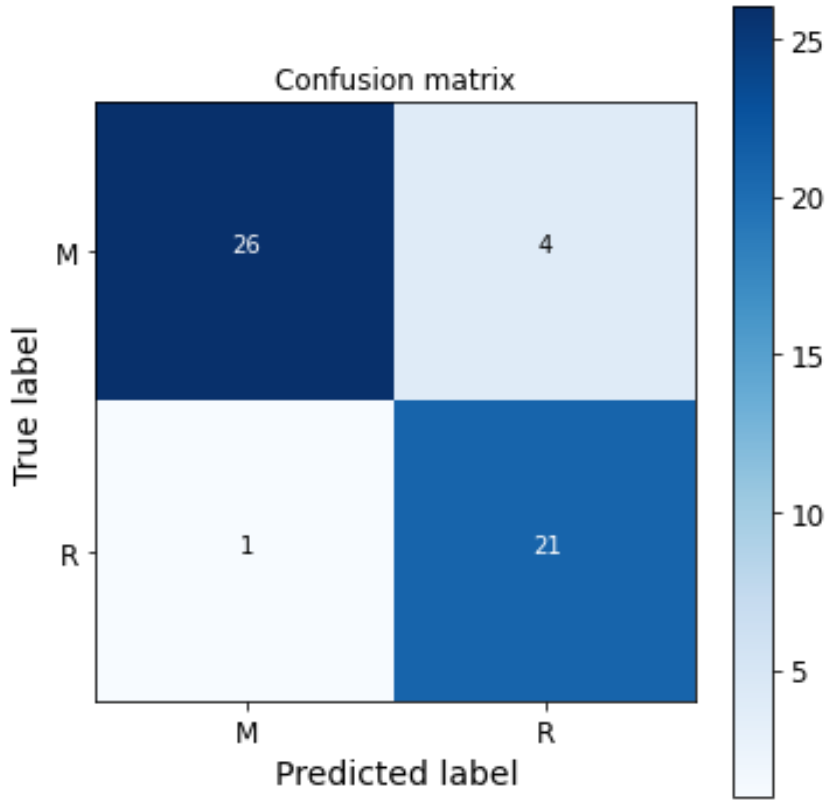
شکل ۲ـ۱۸) Confusion Matrix برای داده تست

2) شبکه با سه لایه ی مخفی :

در این حالت با توجه به افزایش پارامترهای شبکه در صورت مدل دهی مناسب دقت ی تواند کمی افزایش پیدا کند اما با توجه به هزینه ی محاسباتی زیاد بنظر می رسد که در این قسمت این کار بهینه نباشد، شکل مربوط به Confussion Matrix آن در ادامه آورده شده است.هم چنین مطابق شکل ۲ـ۱۹ مشاهده می شود که خیلی زود اوورفیت شده است.



شکل ۲ـ۱۹) دقت و loss به ازای هر ایپاک batch size=32



شکل ۲ـ۲۰) Confusion Matrix برای داده تست

ک) بهترین شبکه در شبکه های بررسی شده فوق با معیار Accuracy را شبکه با سه لایه مخفی مرتبط با شکل ۲ـ۲۰ و ۲ـ۱۹ می توان گفت اما با توجه به افزایش پارامتر ها همان شبکه دو لایه ی بیان شده با batch_size =64 می تواند به صرفه تر باشد. برای عملکرد بهتر شبکه و دقت بالاتر علی رغم اینکه هم چنان می توان با پارامترهای شبکه باز هم در جهت بهینگی بازی کرد اما راه حل مناسب تر پیدا کردن داده با ابعاد مناسب تر برای دادن به ورودی شبکه ء از بین بردن correlation و ... اقدام نمود. در ادامه پارامترهای منتسب به شبکه آورده شده است.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_30 (Dense)             (None, 20)                1220
_________________________________________________________________
dense_31 (Dense)             (None, 256)               5376
_________________________________________________________________
dense_32 (Dense)             (None, 1)                 257
=================================================================
Total params: 6,853
Trainable params: 6,853
Non-trainable params: 0
_________________________________________________________________
```

شکل ۲ـ۲۱) ساختار نهایی انتخاب شده با دو لایه ی نهان

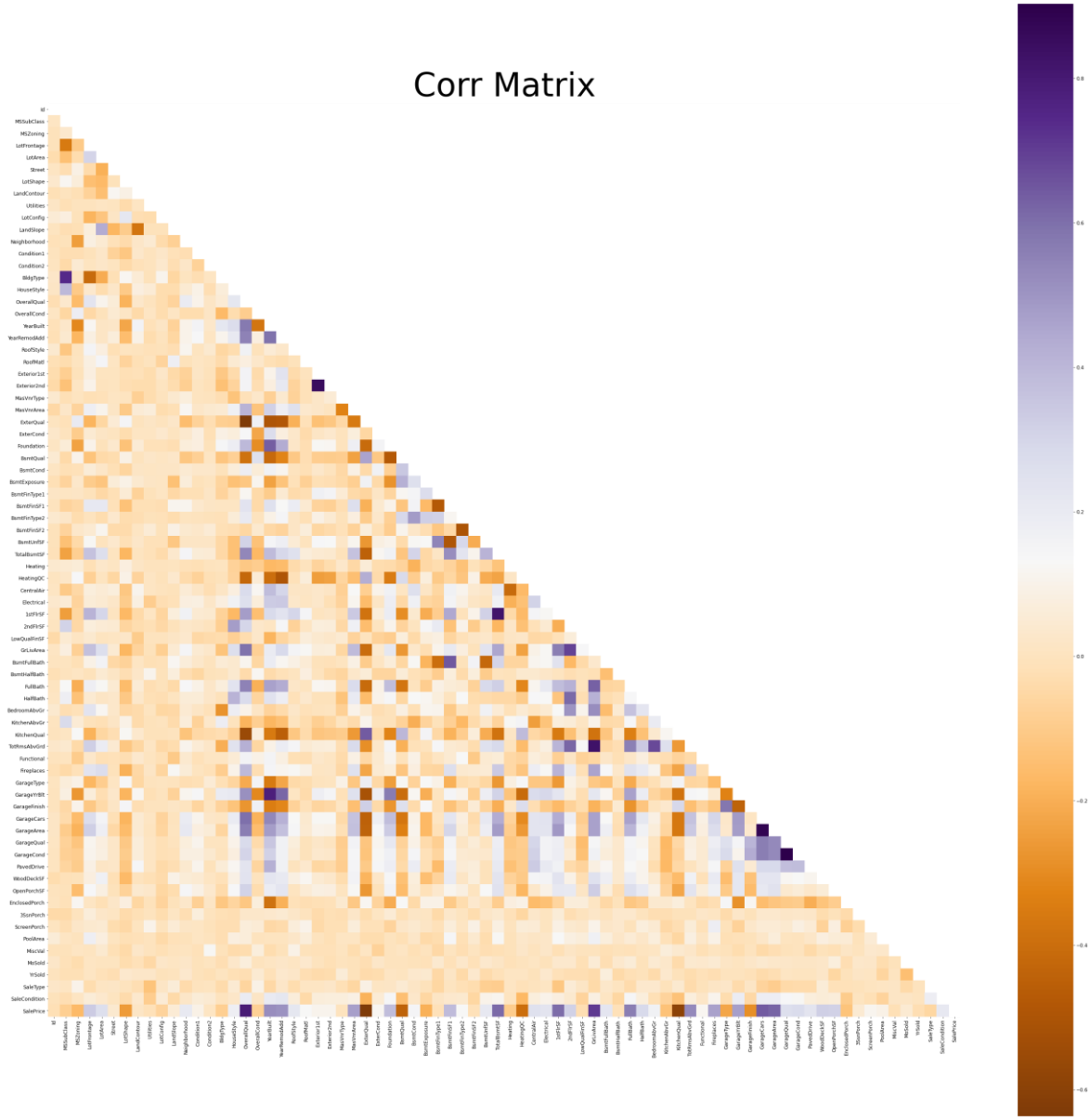
در این شبکه تابع فعالساز های میانی Relu بوده و در لایه ی انتهایی یک sihmoid با حد آستانه ۰,۵ قرار گرفته است.

ل)توضیحات زیر با تجه به نتایج بیان شده در قسمت ک آورده شده است.

با توجه به شکل های ۲ـ۱۷ تا ۲ـ۲۰ مشاهده می گردد که برای شبکه ی درای تعداد نورون بالا و لایه های زیادتر اوورفیت در ایپاک کمتر صورت می گیرد و این فرضیه غلط می باشد.
علت این موضوع را با توجه به آپدیت شدن وزن ها با توجه به قاعد گرادیان شبکه دارای هایپرپارامترهای زیاد همگرایی بیشتری دارد و در ایپاک پایین تر به مقدار بهینه ی خود دست پیدا می کند.

سوال 3 – Dimension Reduction

الف) با توجه به بزرگ بودن دیتا برای رزولوشن بهتر است به نوتبوک این قسمت مراجعه گردد.*



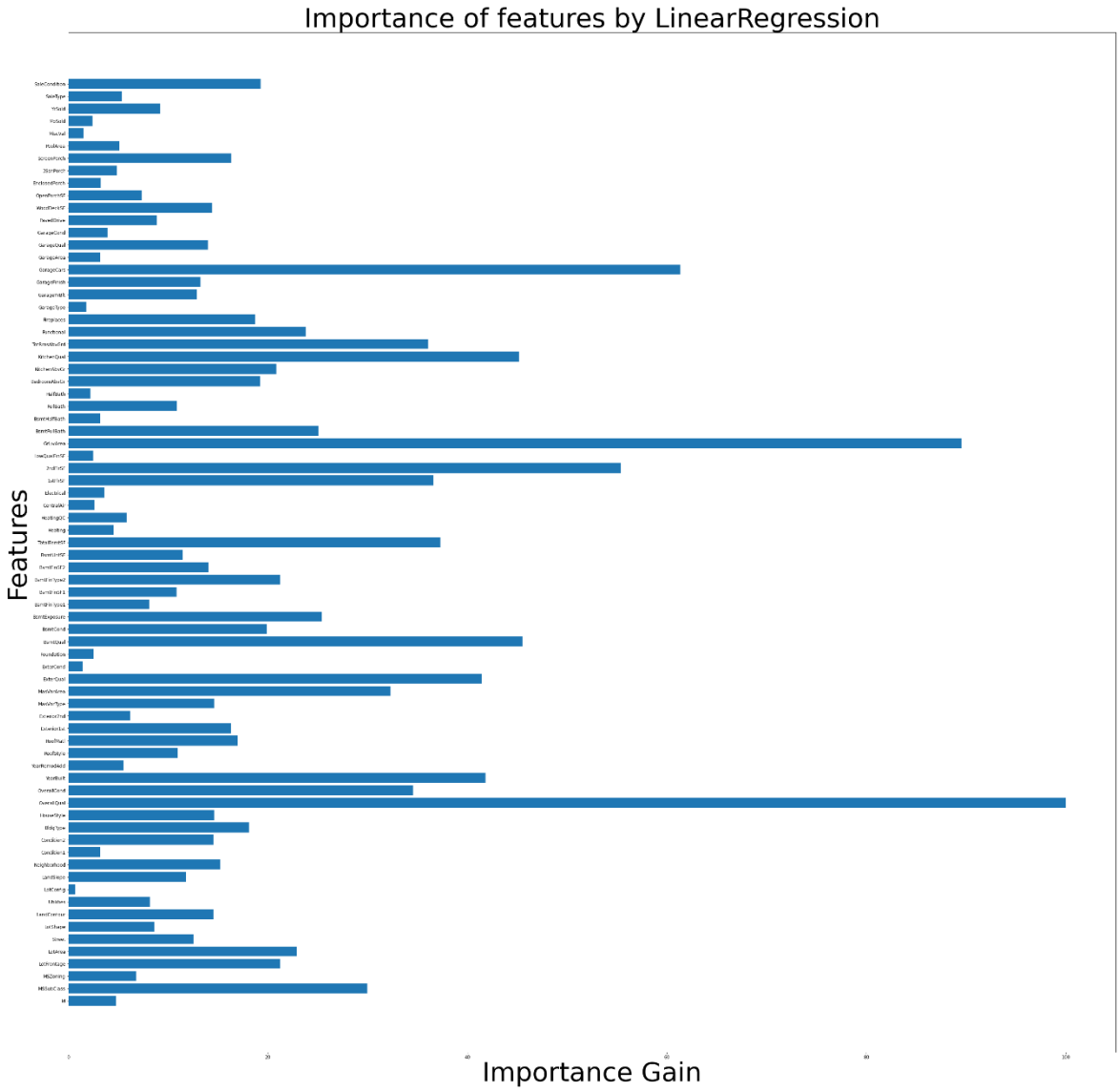
شکل ۳ـ۱) ماتریس کواریانس ویژگی ها

با تجه به شکل فوق ماتریس همبستگی نشان می دهد که دو feature در نظر گرفته شده چه ارتباطی (از رنج ۰ تا ۱) با یک دیگر دارند که نزدیکی به یک نشان دهنده ی ارتباط زیاد بین آن دو بعد می باشد و

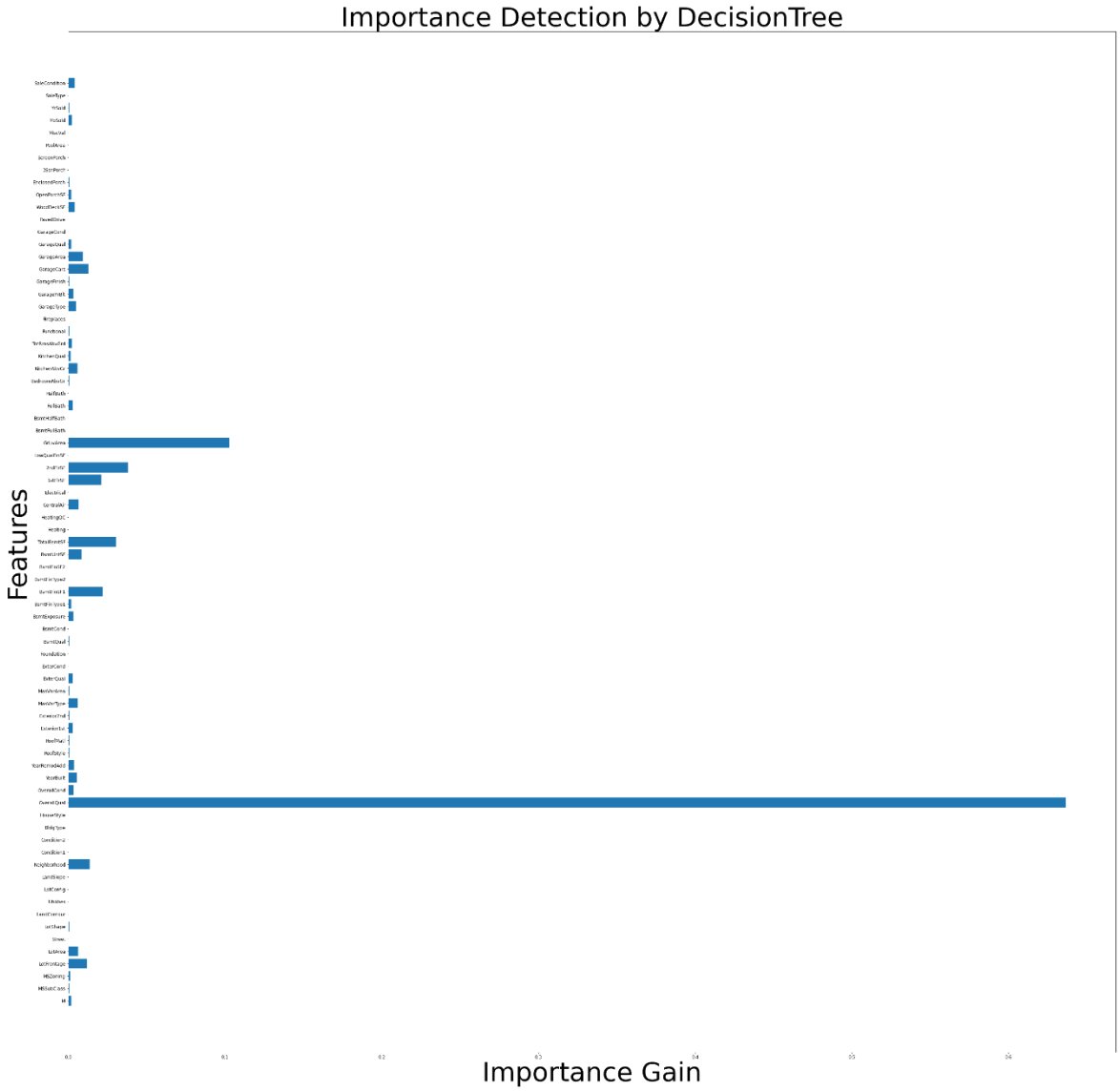
می توان یکی از آن دو را برای مثال در نظر نگرفت.هم چنین طبیعتا بر روی قطر اصلی همواره یک داریم که نشان دهنده ی correlation هر بعد از فضای ویژگی با خودش می باشد. هم چنین با توجه به تقارن کامل ماتریس کواریانس با توجه به تعریف صورت گرفته تنها نیمی از آن ماتریس توسط heatmap فوق صورت گرفته است.

ب **) (امتیازی)

1) بار پلات اول اهمیت هر ویژگی را بطور نرمالایز شده با توجه به الگوریتم Linear Regressionموجود در scikit_Learnنمایش می دهد.



شکل ۳ـ۲) بررسی اهمیت هر ویژگی توسط Linear Regression



شکل ۳ـ۳) اهمیت هر ویژگی به وسیله ی الگوریتم Decision Tree Cart

با توجه به نمودارهای فوق می توان ویژگی های پر اهمیت تر را با قرار دادن یک حد آستانه و بعد از حذف Correlation میان آن ها ، انتخاب نمود.

ج) در روش Backwrd Elimination ابتدا تمامی ویژگی ها را در نظر گرفته سپس با در نظر گرفتن یک حد آستانه 0.05 می توان p_value آن ها را سنجید و بیش ترین مقدار بزرگتر از این حد آستانه را ویژگی مرتبط با آن را حذ می کنیم و این کار را در هر مرحله و تا حذف تمامی ویژگی هیا بزرگتر از این آستانه ادامه می دهیم، زمان و ویژگی حذف شده در هر مرحله در ادامه آورده شده است:

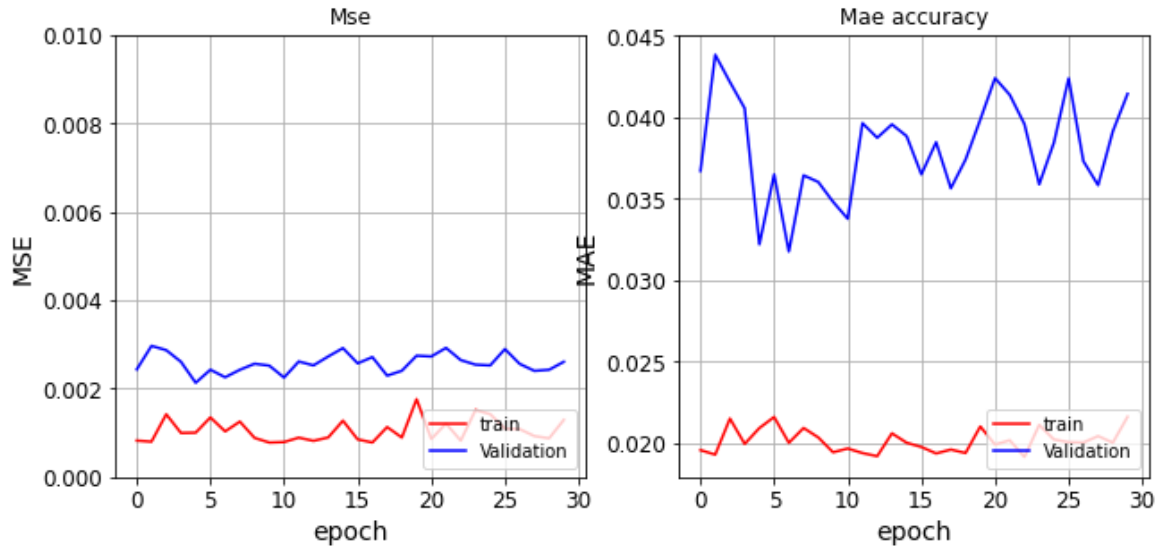
```
Step 01 Execution Time is : 0.025 seconds --- LowQualFinSF
Step 02 Execution Time is : 0.030 seconds --- LotConfig
Step 03 Execution Time is : 0.031 seconds --- BsmtUnfSF
Step 04 Execution Time is : 0.028 seconds --- GarageArea
Step 05 Execution Time is : 0.028 seconds --- ExterCond
Step 06 Execution Time is : 0.028 seconds --- GarageType
Step 07 Execution Time is : 0.026 seconds --- GarageCond
Step 08 Execution Time is : 0.032 seconds --- MiscVal
Step 09 Execution Time is : 0.025 seconds --- Foundation
Step 10 Execution Time is : 0.027 seconds --- HalfBath
Step 11 Execution Time is : 0.025 seconds --- CentralAir
Step 12 Execution Time is : 0.024 seconds --- MoSold
Step 13 Execution Time is : 0.028 seconds --- BsmtHalfBath
Step 14 Execution Time is : 0.025 seconds --- Electrical
Step 15 Execution Time is : 0.023 seconds --- YearRemodAdd
Step 16 Execution Time is : 0.028 seconds --- Exterior2nd
Step 17 Execution Time is : 0.024 seconds --- EnclosedPorch
Step 18 Execution Time is : 0.026 seconds --- Condition1
Step 19 Execution Time is : 0.024 seconds --- SaleType
Step 20 Execution Time is : 0.021 seconds --- 3SsnPorch
Step 21 Execution Time is : 0.025 seconds --- Heating
Step 22 Execution Time is : 0.023 seconds --- PoolArea
Step 23 Execution Time is : 0.021 seconds --- GrLivArea
Step 24 Execution Time is : 0.020 seconds --- GarageYrBlt
Step 25 Execution Time is : 0.021 seconds --- BsmtFinType1
Step 26 Execution Time is : 0.023 seconds --- HeatingQC
Step 27 Execution Time is : 0.021 seconds --- Id
Step 28 Execution Time is : 0.019 seconds --- Utilities
Step 29 Execution Time is : 0.020 seconds --- MSZoning
Step 30 Execution Time is : 0.019 seconds --- OpenPorchSF
Step 31 Execution Time is : 0.018 seconds --- FullBath
Step 32 Execution Time is : 0.019 seconds --- LotShape
Step 33 Execution Time is : 0.019 seconds --- PavedDrive
Step 34 Execution Time is : 0.020 seconds --- YrSold
Step 35 Execution Time is : 0.022 seconds --- GarageFinish
Step 36 Execution Time is : 0.026 seconds --- TotalBsmtSF
Step 37 Execution Time is : 0.017 seconds --- RoofStyle
Step 38 Execution Time is : 0.017 seconds --- BldgType
Step 39 Execution Time is : 0.025 seconds --- Exterior1st
Step 40 Execution Time is : 0.015 seconds --- HouseStyle
```

در نهایت ویژگی های باقی مانده به صورت زیر می شوند که به کمک آن dataframe جدیدی برای ویژگی ها می سازیم لازم به ذکر است Id نیز به ستون اول آن به منظور استفاده در پلات ها آورده شده است.

```
significant Features are  , ['MSSubClass', 'LotFrontage', 'LotArea',
'Street', 'LandContour', 'LandSlope', 'Neighborhood', 'Condition2',
'OverallQual', 'OverallCond', 'YearBuilt', 'RoofMatl', 'MasVnrType',
'MasVnrArea', 'ExterQual', 'BsmtQual', 'BsmtCond', 'BsmtExposure',
'BsmtFinSF1', 'BsmtFinType2', 'BsmtFinSF2', '1stFlrSF', '2ndFlrSF',
'BsmtFullBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'GarageCars',
'GarageQual', 'WoodDeckSF', 'ScreenPorch', 'SaleCondition']
```

۳۶ ویژگی فوق باقی می مانند.

حال مدل را با داده کاهش بعد یافته را با همان معماری قبلی آموزش می دهیم نتایج زیر بدست می آیند:



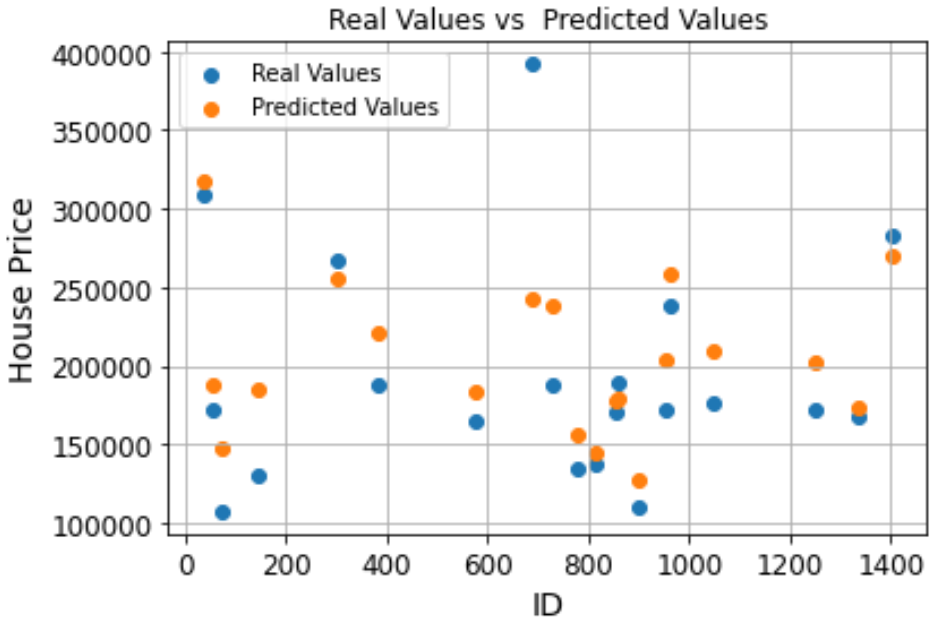
شکل ۳ـ۴) Maeو Mse مربوط به ترین و ولیدیشن

زمان این آموزش 3.3 ثانیه می باشد.

برای داده تست نیز معیارهای زیر بدست می آیند:

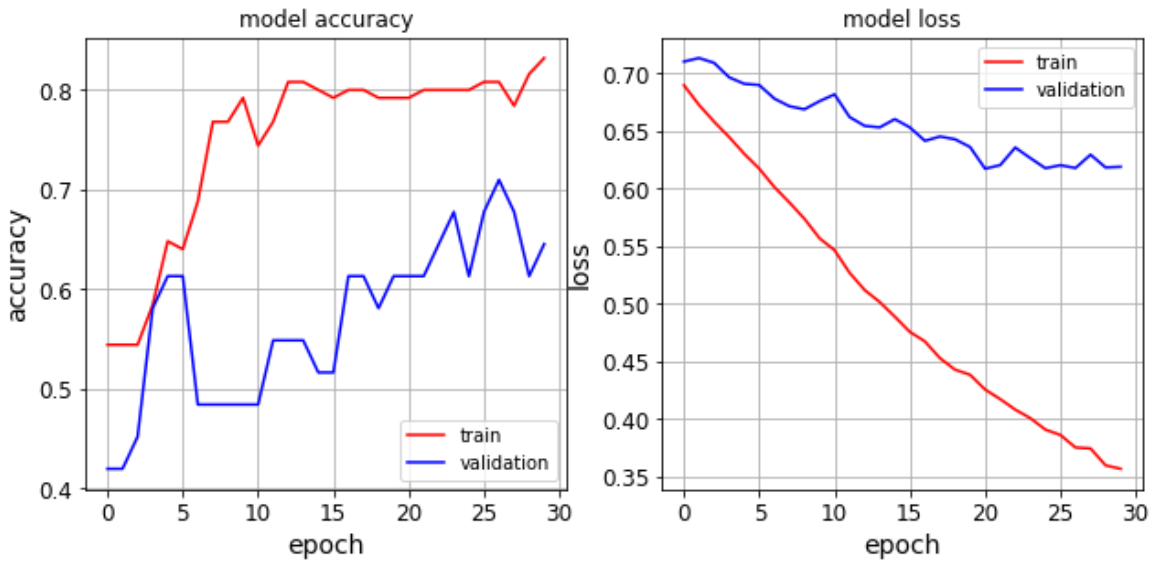
MAE=0.036 , MSE=0.002

شکل نهایی مقادیر واقعی بر حسب مقادیر تخمینی در شکل ۳ـ۵ آورده شده است.



شکل ۳ـ۵) مقادیر واقعی و مقادیر تخمینی بر حسب برخی آیدی های رندوم

د) در این قسمت به کمک متود PCAموجود در کتابخانه SCikit learn ابعاد داده را به ۳۰ کاهش می دهیم .نتیج زیر حاصل می شود.



شکل ۳ـ۶) نمودار های خطا و دقت برای داده ی کاهش مرتبه یافته توسط PCA

زمان انجام این آموزش به شرح زیر می باشد.

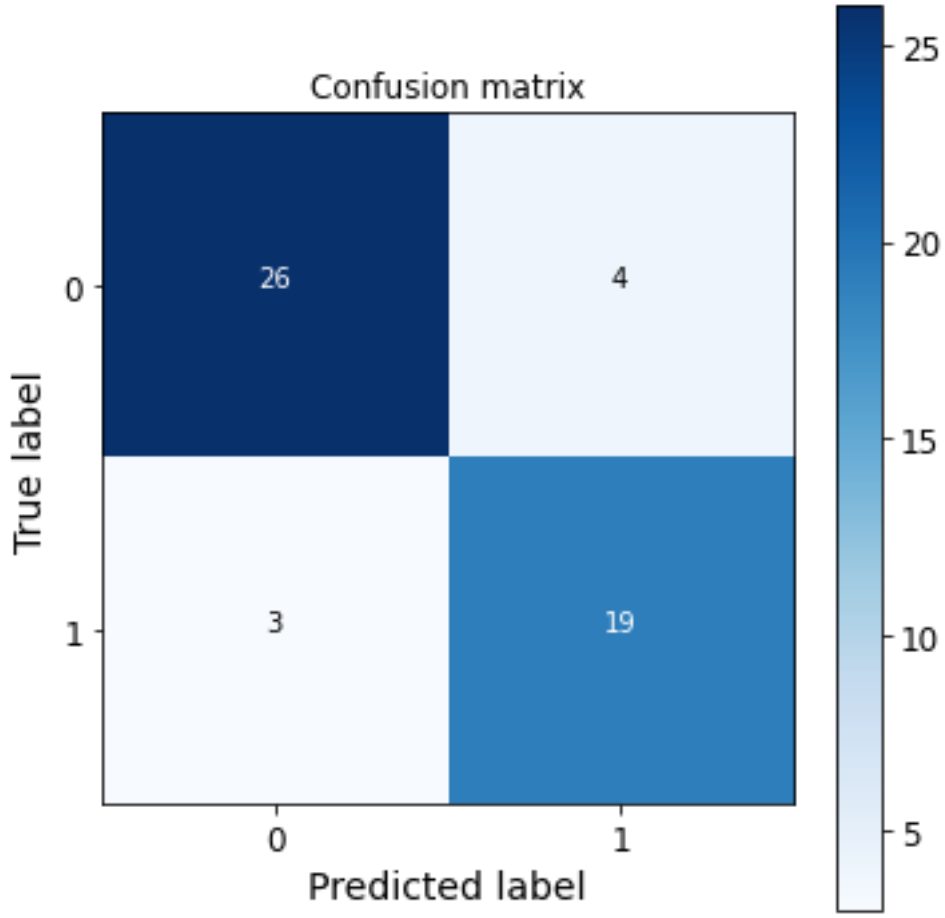
PCA time=0.012 s

Training Time=11.47s

زمان training با توجه به انجام kfold validation زیاد می باشد.

با توجه به داده ی تست نیز به مقادیر زیر دست پیدا می کنیم:

Loss=0.35 , Accuracy=86.54 %



شکل ۳ـ۷) confusion matrix در روش PCA

با توجه به نتایج فوق به دقت بسیار مناسبی در مقایسه با بهترین شبکه بدست آمده در بخش قبلی رسیدیم.

ه)در این بخش از یک آتوانکودر به منظور کاهش ابعاد داده به ۳۰ ویژگی و تست دوباره می پردازیم.

زمان آموزش انکودر=2.78 ثانیه

زمان آموزش شبکه=12.25 ثانیه

هم چنین دقت و خطای مدل نیز بصورت

Acurracy=85.71 % , Loss=0.54

می گردد.

و)

	دقت داده تست	خطای داده تست	زمان
بهترین شبکه سوال ۲	84.46%	0.38	10.43s
AutoEncoder	85.71٪	0.54	15.03 s
PCA	86/54%	0.35	11.48s

جدول ۳ـ۱)

همان طور که از جدول ۳ـ۱ پیداست روش PCAبا کاهش ابعاد داده توانسته به خوبی دقت و خطای متناسب با بهترین شبکه را ایجاد کند در حالیکه زمان زیادی را صرف نکرده است و از نظر مموری لازم برای ذخیره داده ها بهینه است، در اینجا روش AutoEncoder دقت مناسبی دارد که البته با در نظر گرفتن تغییر پارامترها کمی توان عملکرد آن را نیز بهبود بیخشید اما مشاهده می شود که زمان به نسبه زیادی را صرف می کند.در نهایت در راهکار بدون Dimension Reduction بیان شده نیز علی رغم عملکرد مناسب نیاز به حجم کافی برای ذخیره داده ها داریم که بهینه نیست(گرچه در اینجا به علت کوچک بودن حجم داده شاید زیاد به چشم نیاید.)در مجموع و با توصیفات صورت گرفته بنظر می رسد که PCA در این مثال بهترین عملکرد را دارد.