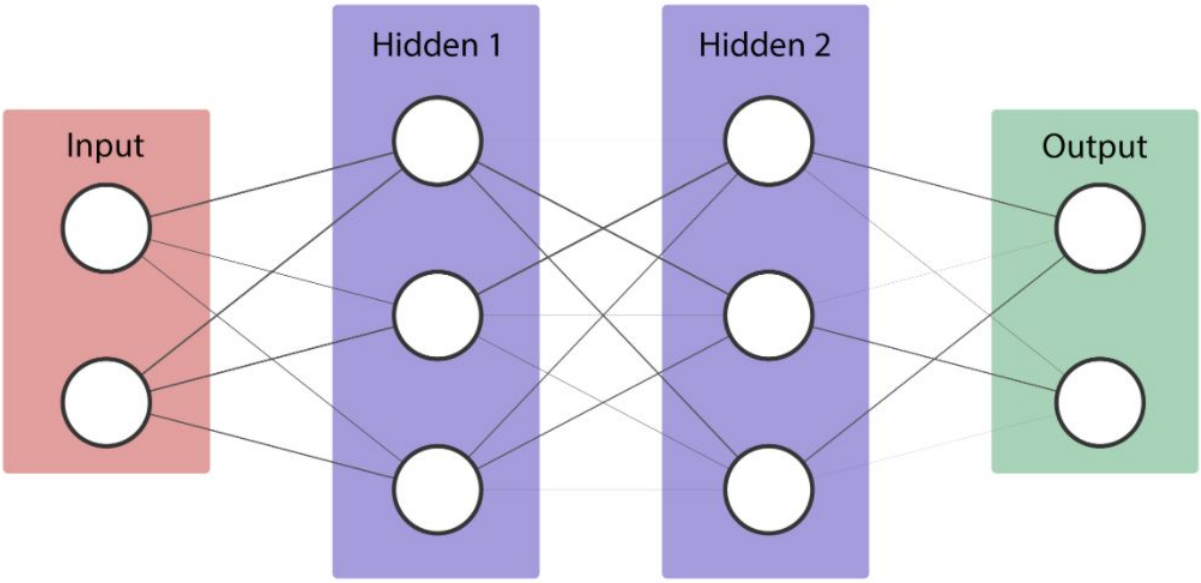
گزارش پروژه اول یادگیری عمیق (MLP)

9 آبان ماه 1399

غزاله محمودی

96522249



نکات :

در این پروژه از کد تمرینات درس هوش محاسباتی نیز استفاده کردم.

به علت زیادی پارامتر ها و تحلیل هایی که برای شبکه های عصبی قابل بررسی است حجم داک اندکی بیش از حد تصور بنده شد. البته بیشتر شامل نتایج و تصاویر آموزش مدل ها می باشد. البته نتایج تصویری مفصل در فایل اجرای کد ها موجود می باشد.

منابع :

Deep Neural Network With L - Layers

Building Autoencoders in Keras

Simple MNIST convnet

How to use K-fold Cross Validation with Keras? – MachineCurve

Implement a neural network from scratch with Python/Numpy — Backpropagation

لینک کدهای اجرایی:

Mlp : section 1, 2, 3

https://colab.research.google.com/drive/1mT4q9HM4oVRI3HVHITPf1ugqWgX8W8h8?usp=sharing

Keras : section 1, 2, 3

https://colab.research.google.com/drive/1mT4q9HM4oVRI3HVHITPf1ugqWgX8W8h8?usp=sharing

Section 4:

https://colab.research.google.com/drive/1i_T9xGAg7b92vvWUcI2sMP97_g3zRmwj?usp=sharing

Section 5:

https://colab.research.google.com/drive/1oZEbi6WkaITR4Txo1KqmE9IrU2qE71N1?usp=sharing

بخش دوم (شبکه MLP)

در این بخش هدف این است که به کمک مدل mlp تسک function approximation را انجام دهیم.
در این قسمت 3 تابع

- sin(x)
- x^2
- X + sin(x)

را مورد بررسی قرار می دهیم.

نتایج :

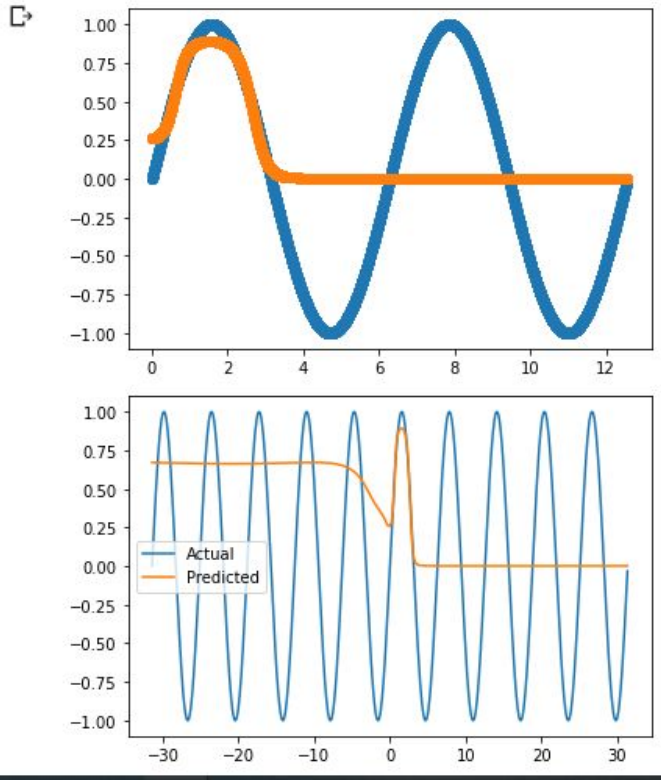
- به کمک این شبکه می توان به تقریب متوسطی از تابع رسید.
- سرعت ران شدن به نسبت پایین است.
- اگر ورودی ها بزرگ باشند وزن ها NAN می شوند و مدل دیگر قادر به رفتار صحیح نیست.
- در کل مدل ساخته شده به کمک این شبکه ضعیف تر به نظر میرسد.
- این جا فقط از یک نوع activation function استفاده کردم. استفاده از activation function متفاوت با توجه به تسک به بهبود شبکه کمک می کند.
- قابلیت تعمیم بسیار پایینی دارد. به عنوان مثال اگر در بازه -1و 1 آموزش ببیند در بازه -2 و 2 بعضی جاها واقعا خوب نیست.
- درباره تابع sin بیرون از بازه آموزش دیده بسیار بسیار بد عمل میکند.
- البته در خوده بازه ترین شده و با داده های آموزشی هم گاهی با تابع اصلی تفاوت دارد.(که این تفاوت در X^2 به خوبی قابل ملاحظه است)
- برای ترین مناسب ورودی باید به اندازه کافی باشد.
- تابع X + sin(x) را با کیفیت بسیار پایین آموزش دید. بزرگتر کردن شبکه تاثیر کمی در بهبود دارد.

نتایج تصویری :

(کلیه نتایج به تفضیل در فایل مربوط به قسمت 1 پروژه قابل مشاهده است و در ادامه بخش اندکی از نتایج آورده شده است)

Same as Keras

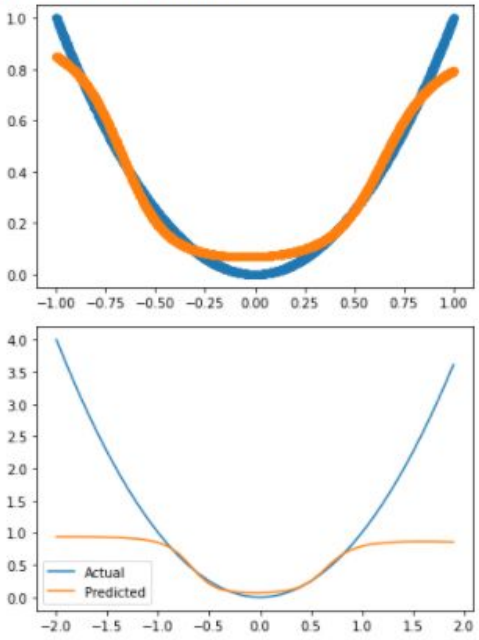
```
x = 4*np.pi*np.random.rand(10000).reshape(1, -1)
y = np.sin(x)

x_test = np.arange(-10*np.pi, 10*np.pi, 0.1).reshape(1, -1)
y_test = np.sin(x_test)

layer = [1, 128, 64, 16, 1]
activations =['sigmoid', 'sigmoid', 'sigmoid', 'sigmoid']
main(layer , activations, x, y, x_test, y_test, 1000, 0.6)
```



same as keras

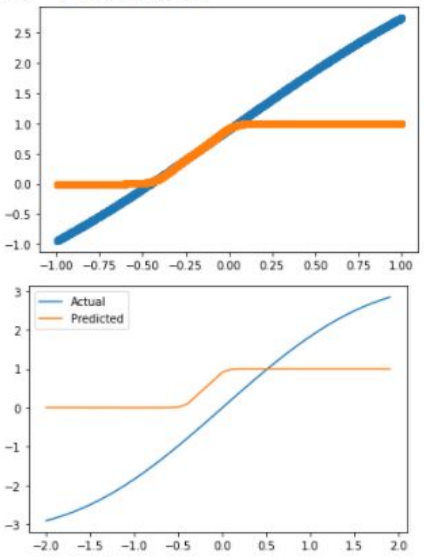
```
x = np.random.uniform(-1, 1, 2000).reshape(1, -1)
y = x**2

x_test = np.arange(-2, 2, 0.1).reshape(1, -1)
y_test = x_test**2

layer = [1, 128, 64, 16, 1]

activations =['sigmoid', 'sigmoid', 'sigmoid', 'sigmoid']

main(layer , activations, x, y, x_test, y_test, 1000, 0.6)
```

```
loss = 2.3601912158791287
```



```
x = np.random.uniform(-1, 1, 2000).reshape(1, -1)
y = x + np.sin(x) + np.random.uniform(low=0.9, high=0.9, size=(len(x)))

x_test = np.arange(-2, 2, 0.1).reshape(1, -1)
y_test = x_test + np.sin(x_test)

layer = [1, 512, 255, 128, 1]
activations =['sigmoid', 'sigmoid', 'sigmoid', 'sigmoid']

main(layer , activations, x, y, x_test, y_test, 10000, 0.5)
```

```
loss = 34.99607615379013
```



بخش دوم (شبکه MLP آماده keras)

https://colab.research.google.com/drive/17pMrARQMggyRrUd6jSQE1m0L_iD1x4hs?usp=sharing

در این بخش هدف این است که به کمک مدل mlp تسک function approximation را انجام دهیم.
در این قسمت 3 تابع

- sin(x)
- x^2
- X + sin(x)

را مورد بررسی قرار می دهیم.

```
# Define model
model = Sequential()
model.add(Dense(128, input_dim=1, activation='relu'))
model.add(Dropout(0.02))
model.add(Dense(64, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
history = model.fit(x_train, y_train, epochs=epoch, batch_size=512, verbose = 0,
                    validation_data = (x_test, y_test))
```

نتایج :

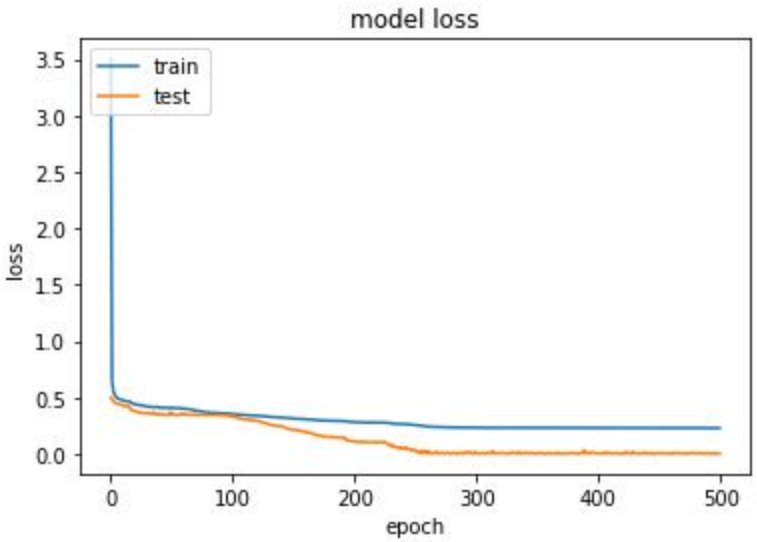
- هر چه بازه اعداد ورودی بیشتر باشد، یادگیری تابع برای شبکه mlp آسان تر است.
- این شبکه تقریب خوبی از توابع را به ما میدهد.
- در تابع X^2 با ثابت نگه داشتن بازه ورودی و تنها افزایش تعداد epoch ها بهبود خوبی در تقریب تابع مشاهده کردم. بازه ورودی بین -10 و 10 بود و خروجی ما بین -55 و 55 به خوبی تقریب زده شد. این تابع از لحاظ سختی تابع متوسطی می باشد و با تعداد ورودی مناسب می توان به خوبی آن را تقریب زد.
- برای مثال برای یادگیری صحیح تابع سینوس به تعداد داده آموزشی بیشتری نسبت به تابع X^2 نیاز می باشد.

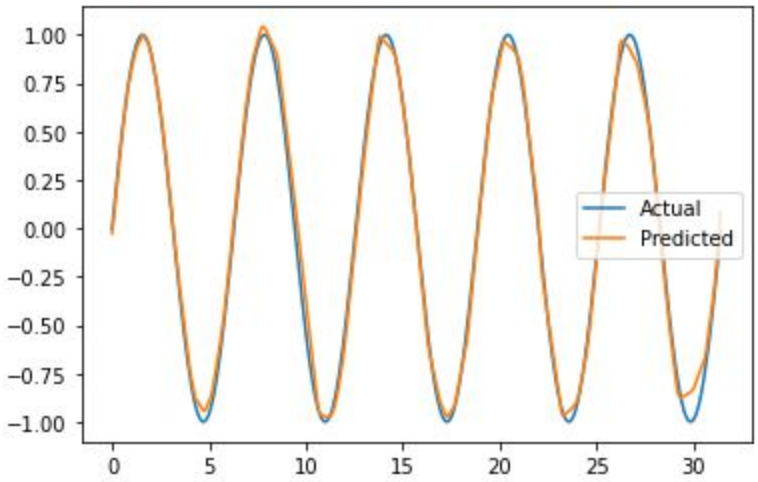
- با بررسی های انجام شده دریافتم برای تابعی مثل سینوس تعداد داده های آموزشی موثر تر از عمیق تر بودن شبکه و افزایش نورون های هر لایه می باشد.
- در سینوس به طور واضحی به تعداد داده بیشتری نیاز داریم. به نظر میرسید تاثیر داده در یادگیری بیشتر از افزایش بیهوده تعداد epoch ها تاثیر گذار است.
- به نسبت درجه سختی تابع شبکه بزرگ تر و تعداد epoch بیشتر موجب بهبود کیفیت یادگیری می شود.
- گرچه طبق تجربه قبلی بنده برای تقریب تابع شبکه هایی مثل RBF کاربرد بهتری به نسبت mlp دارند.
- نکته ای که همواره باید مورد توجه قرار داد این است که تعداد epoch و لایه ها و نورون های هر لایه را به گونه ای انتخاب کنیم که علاوه بر آموزش صحیح و کامل مانع overfit شویم.
- انتخاب صحیح activation function هم در بهبود عملکرد بسیار کمک کننده است.

نتایج تصویری :

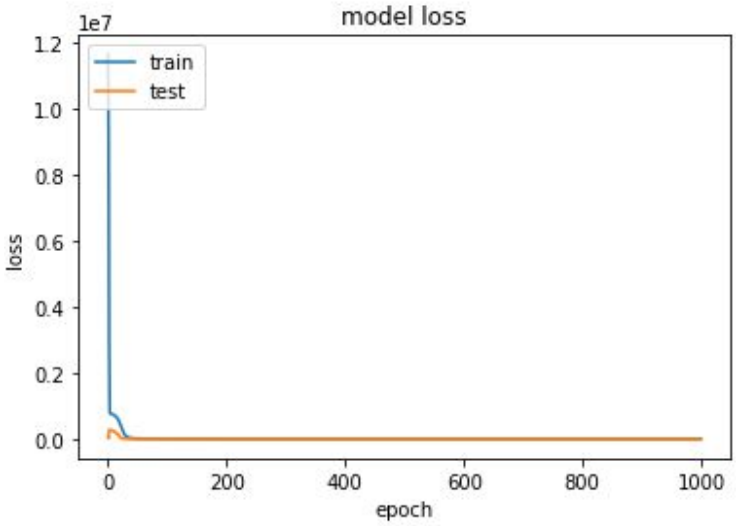
(کلیه نتایج به تفضیل در فایل مربوط به قسمت 3 پروژه قابل مشاهده است و در ادامه بخش اندکی از نتایج آورده شده است)

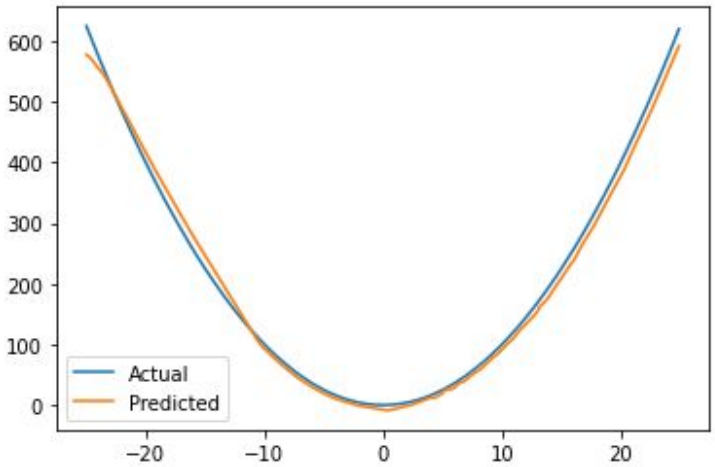
```
x = np.random.random((40000,1))* 100 - 20
y = np.sin(x)

x_test = np.arange(0, 10*np.pi, 0.1)
y_test = np.sin(x_test)

model(x, y, x_test, y_test, 500)
```



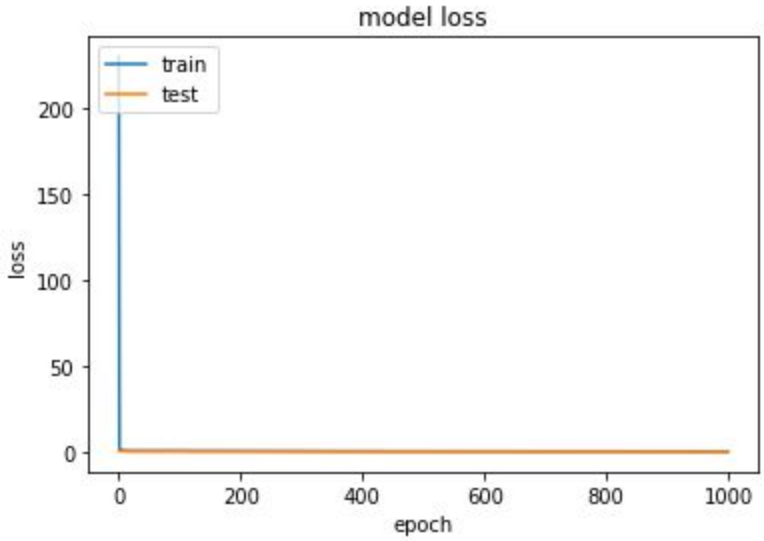


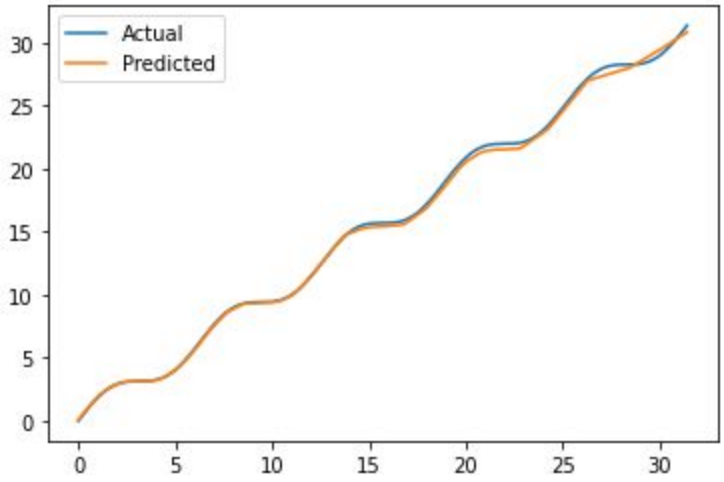
```
# -10 < x < 10
x = np.random.random((25000,1))* 100 - 10
y = x**2

x_test = np.arange(-25, 25, 0.1)
y_test = x_test**2

model(x, y, x_test, y_test,1000)
```





```
x = np.random.random((40000,1))* 100 - 10
y = np.sin(x) + x

x_test = np.arange(0, 10*np.pi, 0.1)
y_test = np.sin(x_test) + x_test

model(x, y, x_test, y_test, 1000)
```





بخش دوم (مقایسه دو شبکه)

نتایج :

- در شبکه آماده keras می توانستم بازه اعداد ورودی را بزرگ بدم و به مشکلی بر نمی خوردم. اما در شبکه خودم در بازه های بزرگ وزن ها از جایی به بعد NAN می شد و آموزش متوقف میشد.
- به طور کلی افزایش لایه ها یا نورون های هر لایه یا epoch ها در هر دو موجب بهبود نتایج می شود ولی تاثیر این افزایش مقادیر در شبکه keras بیشتر است.
- به نظر می رسد قدرت یادگیری شبکه keras بیشتر است و با epoch کمتر و اندازه شبکه کوچک نر قادر به یادگیری بهتر می باشد.
- به نظر می آید با فرض مساوی بودن اندازه شبکه ها و epoch شبکه keras با سرعت بیشتری train می شود.
- البته استفاده از activation function به جز سیگموئید با توجه به کاربرد و نوع خروجی ها به کارکرد بهتر در keras کمک می کند.
- قابلیت تعمیم مدل ساخته شده با keras به وضوح بهتر است.
- به نظر میرسید برای آموزش مناسب شبکه keras به تعداد داده کمتری نیاز دارد.
- شبکه keras برای یادگیری توابع پیچیده تر مناسب تر است.

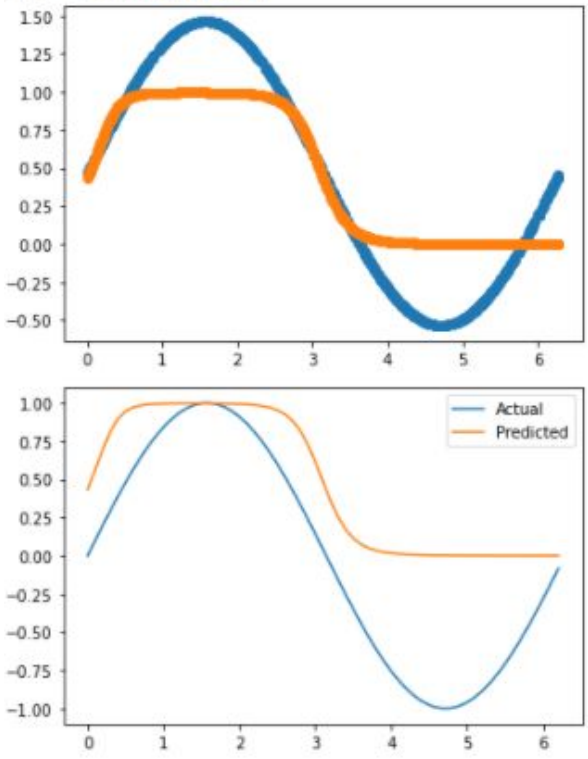
بخش سوم (شبکه MLP)

در این بخش قصد داریم ورودی نویزی را به مدل mlp بدهیم و نتایج را مقایسه کنیم. به این صورت عمل کردم که به عدد خروجی تابع ها مقدار رندومی را اضافه کردیم. برای افزایش نویز بازه عدد رندوم اضافه شده را افزایش دادم.

نتایج تصویری :

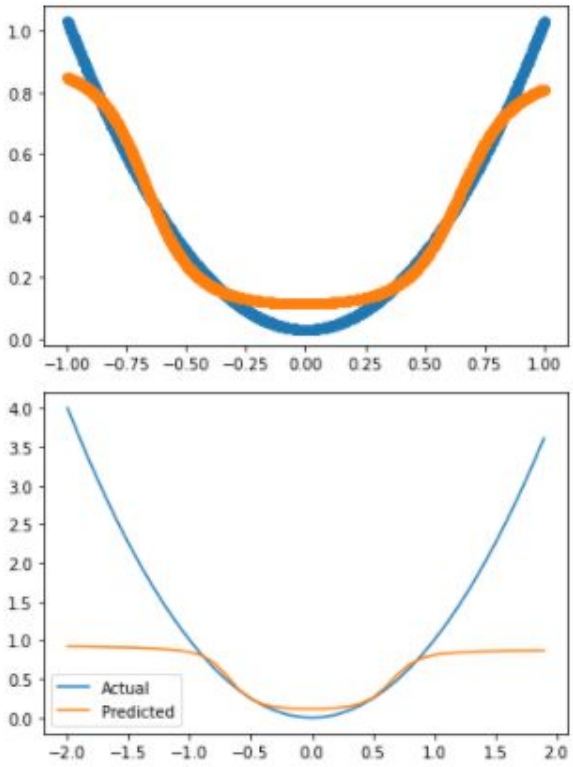
```
x = 2*np.pi*np.random.rand(1000).reshape(1, -1)
y = np.sin(x) + np.random.uniform(low=-0.7, high=0.7, size=(len(x)))

x_test = np.arange(0, 2*np.pi, 0.1).reshape(1, -1)
y_test = np.sin(x_test)

layer = [1, 128, 64, 1]
activations =['sigmoid', 'sigmoid', 'sigmoid']
main(layer , activations, x, y, x_test, y_test, 1000, 0.6)
```

```
.oss = 9.76989066359129
```



```
x = np.random.uniform(-1, 1, 2000).reshape(1, -1)
y = x**2 + np.random.uniform(low=-0.1, high=0.1, size=(len(x)))

x_test = np.arange(-2, 2, 0.1).reshape(1, -1)
y_test = x_test**2

layer = [1, 128, 64, 16, 1]

activations =['sigmoid', 'sigmoid', 'sigmoid', 'sigmoid']

main(layer , activations, x, y, x_test, y_test, 1000, 0.6)
```

```
loss = 2.6230521213097413
```

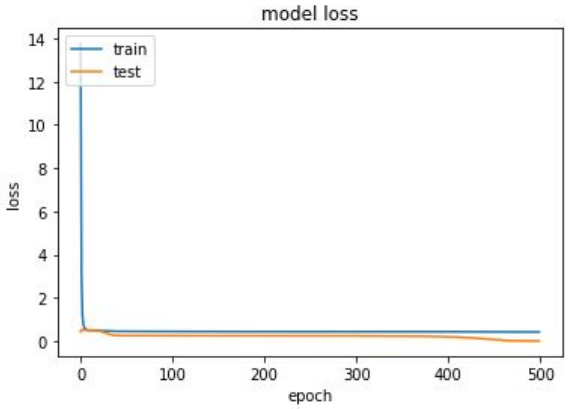


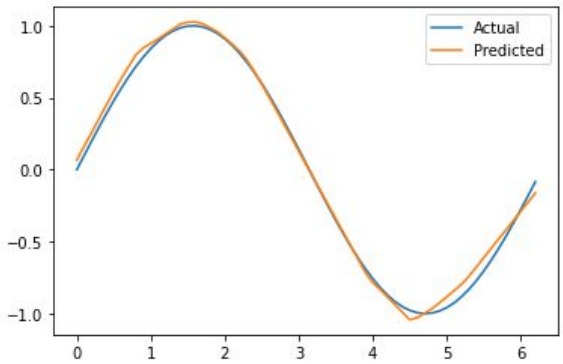
بخش سوم (شبکه MLP آماده keras)

در این بخش قصد داریم ورودی نویزی را به مدل mlp بدهیم و نتایج را مقایسه کنیم. به این صورت عمل کردم که به عدد خروجی تابع ها مقدار رندومی را اضافه کردیم. برای افزایش نویز بازه عدد رندوم اضافه شده را افزایش دادم.

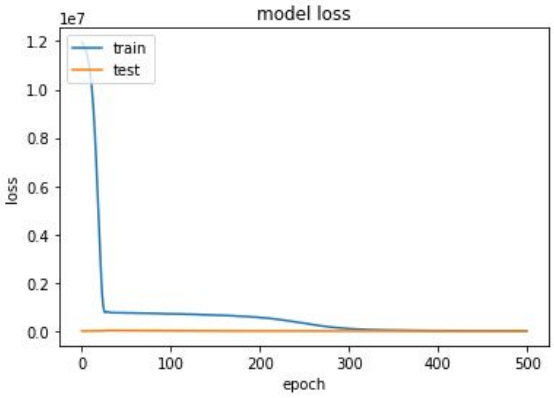
نتایج تصویری :

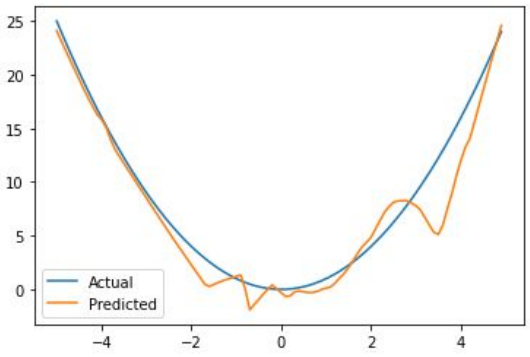
```
x = np.random.random((4000,1))* 100 - 10
y = np.sin(x) + np.random.uniform(low=-0.1, high=0.1, size=(len(x)))

x_test = np.arange(0, 2*np.pi, 0.1)
y_test = np.sin(x_test)

model(x, y, x_test, y_test, 500)
```



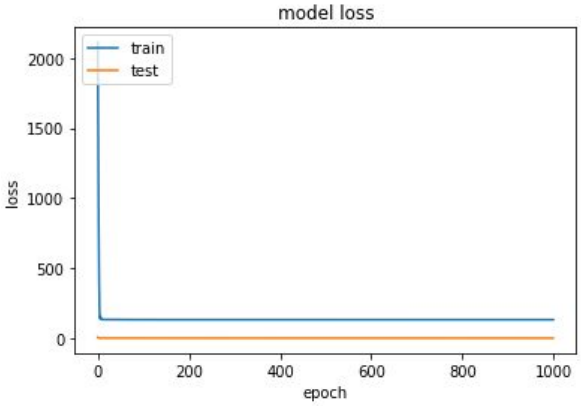


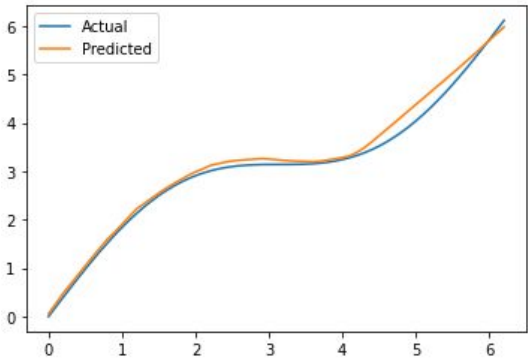
```
x = np.random.random((2500,1))* 100 - 10
y = x**2 + np.random.uniform(low=-50, high=+50, size=(len(x)))

x_test = np.arange(-5, 5, 0.1)
y_test = x_test**2

model(x, y, x_test, y_test ,500)
```





```
x = np.random.random((4000,1))* 100 - 10
y = np.sin(x) + x + np.random.uniform(low=-20, high=20, size=(len(x)))

x_test = np.arange(0, 2*np.pi, 0.1)
y_test = np.sin(x_test) + x_test

model(x, y, x_test, y_test, 1000)
```





بخش سوم (مقایسه دو شبکه)

نتایج :

- در این قسمت به ازای هر تابع 1 تا 3 سطح نویز مختلف را بر روی هر 3 تابع اعمال کردم. نویز بر روی تابع پیچیده تر موثر تر بود و به عنوان مثال کلیت تابع متوسط مثل x^2 کمتر خراب می کرد.
- به نظر میرسد شبکه کراس نسبت به نویز مقاوم تر از mlp خودم می باشد.
- با نویز یکسان خروجی mlp خودم بیشتر خراب می شود.
- نویز بر روی توابع پیچیده تر تاثیر بیشتری میگذارد و خروجی را بیشتر دچار اختلال می کند.
- افزایش نویز در keras نسبت به mlp خودم تاثیر کمتری بر خراب شدن خروجی تابع می گذارد.

بخش چهارم

https://colab.research.google.com/drive/1i_T9xGAg7b92vvWUcI2sMP97_g3zRmwj?usp=sharing

در این بخش قصد داریم با استفاده از ماژول کراس با دیتاست mnist کار کنیم . ابتدا به کمک ماژول های کراس دیتا ست را ذخیره می کنیم .

```
from keras.datasets import mnist
from keras.utils import np_utils, to_categorical
```

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

تصویر ورودی در حال حاضر یک ماتریس 28 * 28 است که آن را reshape کرده و به ماتریس 1 بعدی 784 تایی تبدیل می کنیم.
همچنین با توجه به مواردی که در درس یاد گرفتیم برای اینکه تاثیر همه ورودی ها تقریبا یکسان و در نرمال باشد داده ها نرمال کرده (بر 255 تقسیم میکنیم) تا مقدارشان عددی بین صفر و یک شود.

```
x_train = train_images.reshape((60000, 784))
x_train = x_train.astype('float32')/255

x_test = test_images.reshape((10000, 784))
x_test = x_test.astype('float32')/255
```

تسک ما در واقع classification است. داده های خروجی شامل اعداد بین 0 تا 9 هستند که نشان دهنده کلاس آن داده می باشد. اگر از خوده اعداد 0 تا 9 استفاده کنیم این اختلاف اعداد روی مدل تاثیر می گذارد. بنابراین داده ها را one hot encoding می کنیم تا از این تاثیر جلوگیری کنیم .
و این کد گذاری به صورت مقابل است . برای مثال عدد یک به صورت [0, 0, 0, 0, 0 , 0, 0, 0, 1, 0] کد گذاری می شود.

```
y_train =  to_categorical(train_labels) #one hot encoding
y_test = to_categorical(test_labels)
```

مدل استفاده شده 4 لایه است . لایه ورودی 784 نورون و لایه های hidden به ترتیب 128و 64نورون و لایه خروجی 10 نورون که صفر یا یک بودن هر کدام نشان دهنده بودن یا نبودن عدد مورد نظر است(به طور مثال مقدار ایندکس 1 آرایه از 1 باشد یعنی عدد تشخیصی از نظر مدل یک است و اگر 0 باشد یعنی مقدار تشخیصی یک نیست)
اینکه تعداد لایه های مخفی دو باشد یا بیشتر را با آزمون و خطا امتحان کردم و با 2 لایه نتیجه بهتری میدهد.(با افزایش لایه دقت داشت کمتر میشد و سیر نزولی داشت) همچنین با تجربه یافتم تابع relu برای لایه میانی و softmax برای لایه خروجی بهتر جواب می دهد. همچنین مقادیر مختلفی را برای تعداد نورون ها بررسی کردم که مقادیر ذکر شده بهترین عملکرد و دقت را داشت و مدل را به صورت dense تعریف می کنیم و بدان معنی که کلیه نورون های هر لایه به یکدیگر وصل هستند.(fully connected)
همچنین از اپتیمایزر adam استفاده کردم که طبق سرچ ها گویا جواب بهتری میده و ازش در اجرا الگوریتم کاهش گرادیان استفاده میشود.
داده ها را به دو قسمت train , test تقسیم کردیم. 60000 داده آموزشی و 10000 داده تست داریم.
مدل را توسط k-fold-cross validation آموزش داده و سپس توسط unseen data به بررسی دقیق تر دقت مدل می پردازیم.
حال به بررسی تاثیر پارامتر ها بر دقت و کیفیت شبکه آموزش دیده می پردازیم. در این مرحله تعداد لایه ها و تعداد نورون ها و تعداد epoch ها را تغییر می دهیم و تاثیر تغییرات بر بهبود شبکه را مورد بررسی قرار می دهیم.

نتایج :

- با افزایش نورون ها و لایه ها دقت مدل افزایش می یابد. البته باید دقت داشت افزایش بیش از حد تاثیری در دقت مدل ندارد و تنها باعث overfit می شود.
- با یک شبکه کوچک تا دقت 90 درصد می توان دست یافت اما از این به بعد برای افزایش دقت نیاز باید شبکه را خیلی بزرگتر کنیم تا دقت اندکی بهبود یابد.
- با فرض یکسان بودن شبکه ها در صورتی که تعداد داده ها کاهش یابد دقت هم کاهش می یابد. به نظر می رسد بهتر است که شبکه انواع داده ها را ببیند تا دقت مناسب تری داشته باشد.

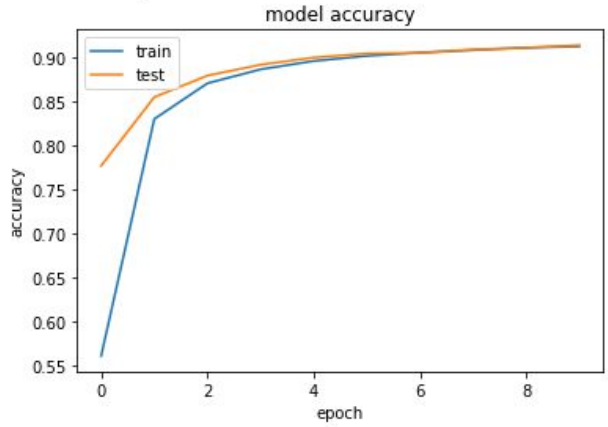
ویژگی های شبکه :

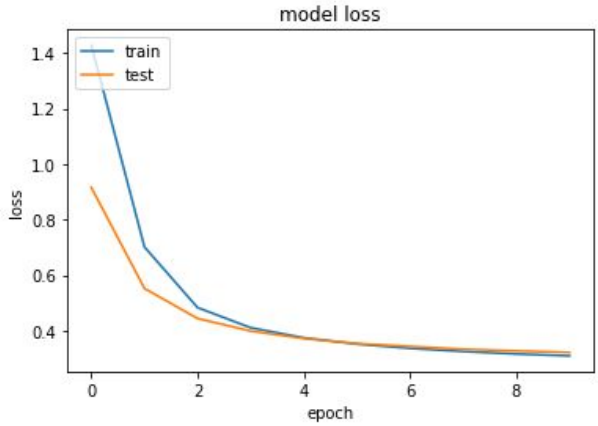
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 1 لایه پنهان 8 نورون
- Fold cross validation 10
- epoch 10

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.3221208453178406 - Accuracy: 90.6166672706604%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.3416956663131714 - Accuracy: 90.23333191871643%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.3452674448490143 - Accuracy: 90.18333554267883%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.33414870500564575 - Accuracy: 90.6333327293396%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.3530931770801544 - Accuracy: 90.39999842643738%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.3286415934562683 - Accuracy: 90.76666831970215%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.34587350487709045 - Accuracy: 89.85000252723694%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.3046182096004486 - Accuracy: 91.20000004768372%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.31864994764328003 - Accuracy: 90.86666703224182%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.32237982749938965 - Accuracy: 91.38333201408386%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 90.61333358287811 (+- 0.44527085590956655)
> Loss: 0.3316488921642303
```

```
Test score: 0.30492180585861206
Test accuracy: 91.46999716758728
```





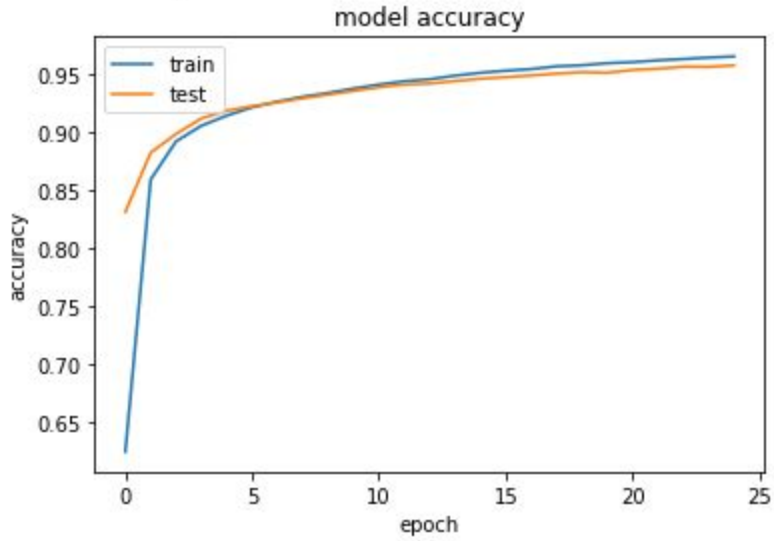
ویژگی های شبکه :

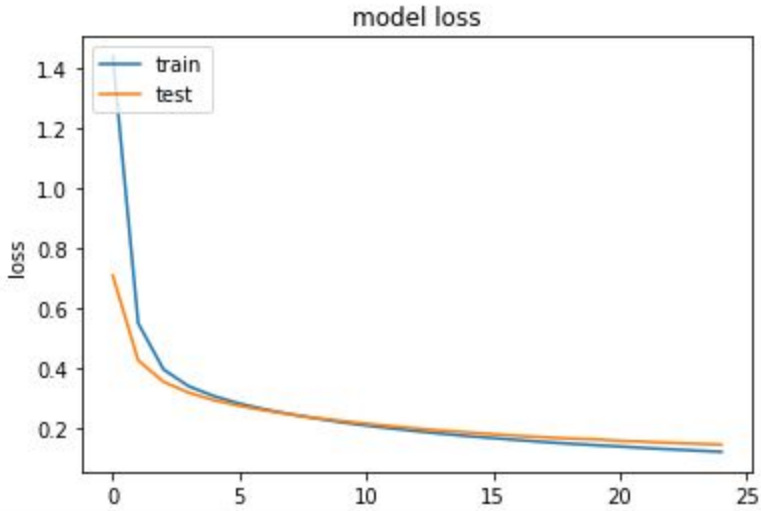
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 1 لایه پنهان با 64 نورون
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.13965998589992523 - Accuracy: 95.99999785423279%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.14880958199501038 - Accuracy: 95.46666741371155%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.15290062129497528 - Accuracy: 95.80000042915344%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.1381286233663559 - Accuracy: 96.11666798591614%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.15657475590705872 - Accuracy: 95.49999833106995%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.15172366797924042 - Accuracy: 95.78333497047424%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.14632190763950348 - Accuracy: 95.6166684627533%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.12763257324695587 - Accuracy: 96.38333320617676%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.14046062529087067 - Accuracy: 95.80000042915344%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.14753247797489166 - Accuracy: 95.78333497047424%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 95.82500040531158 (+- 0.26637993452647407)
> Loss: 0.14497448205947877
```

```
Test loss : 0.13658550381660461
Test accuracy: 95.92000246047974
```



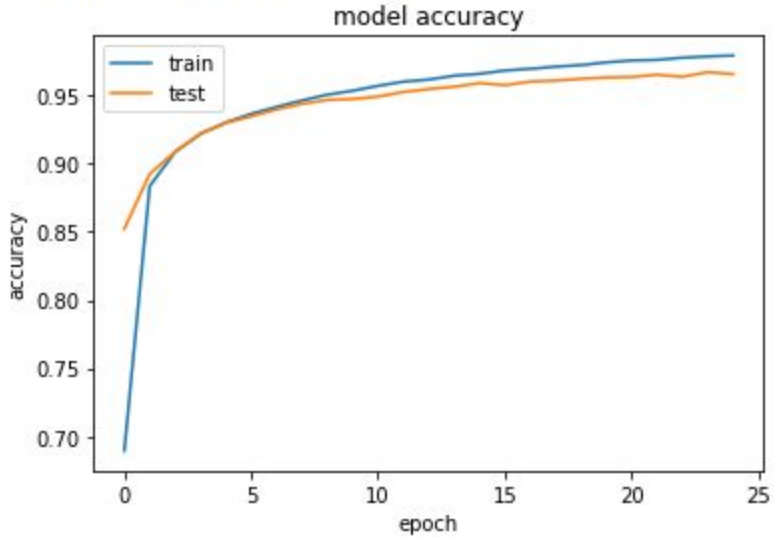


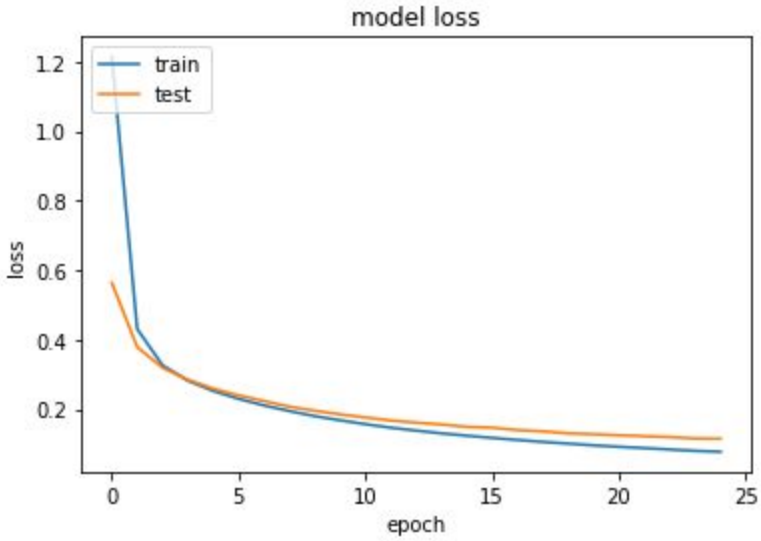
ویژگی های شبکه :

- شبکه با 60000 داده آموزشی و 10000 داده تست
- 1 لایه پنهان و 128 نورون
- Fold cross validation 10
- epoch 25

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.10714292526245117 - Accuracy: 96.74999713897705%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.10796041786670685 - Accuracy: 96.76666855812073%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.11038242280483246 - Accuracy: 96.98333144187927%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.11220848560333252 - Accuracy: 96.79999947547913%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.11387845128774643 - Accuracy: 96.48333191871643%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.10700690001249313 - Accuracy: 96.95000052452087%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.11649208515882492 - Accuracy: 96.79999947547913%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.10996132344007492 - Accuracy: 96.71666622161865%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.12686757743358612 - Accuracy: 96.08333110809326%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.11530587822198868 - Accuracy: 96.5499997138977%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 96.68833255767822 (+- 0.24866911857349072)
> Loss: 0.11272064670920372
```

```
Test loss : 0.10272404551506042
Test accuracy: 96.85999751091003
```



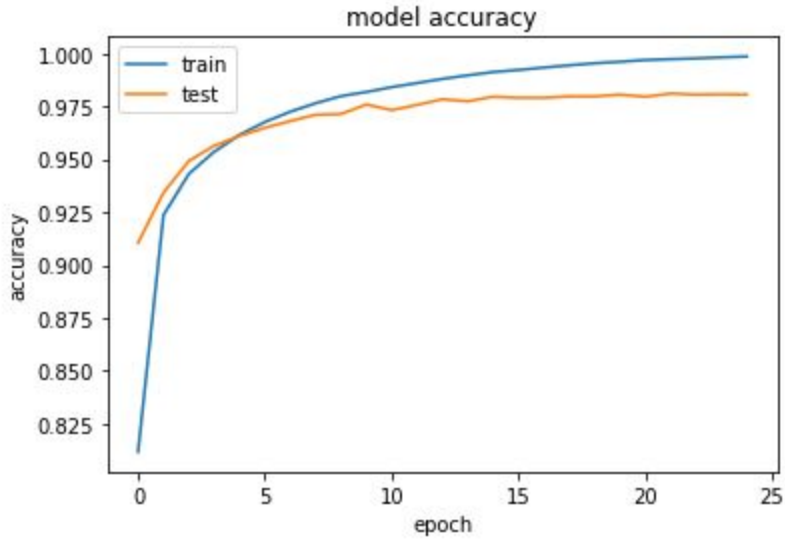


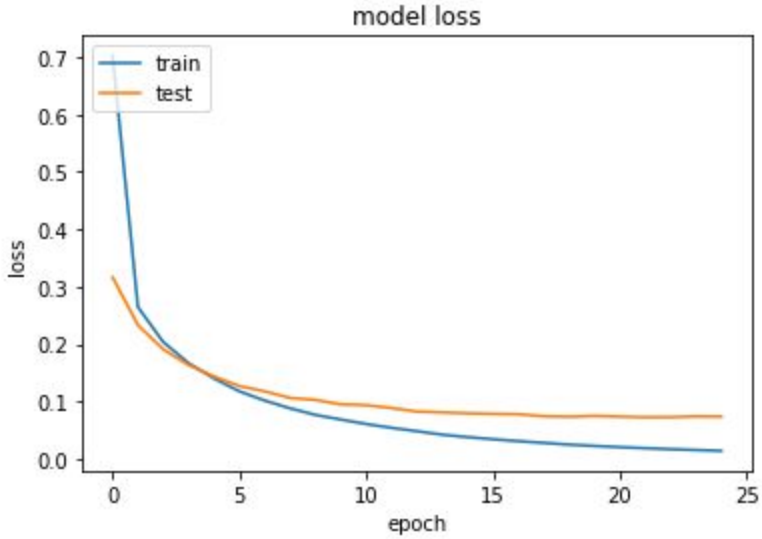
ویژگی های شبکه :

- شبکه با 60000 داده آموزشی و 10000 داده تست
- 1 لایه پنهان با 1024 نورون
- Fold cross validation 10
- epoch 25

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.06671295315027237 - Accuracy: 97.96666502952576%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.06705371290445328 - Accuracy: 98.1166660785675%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.06618962436914444 - Accuracy: 98.15000295639038%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.06778624653816223 - Accuracy: 97.85000085830688%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.058458853513002396 - Accuracy: 98.16666841506958%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.07846178859472275 - Accuracy: 97.60000109672546%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.06883779168128967 - Accuracy: 97.98333048820496%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.0748104453086853 - Accuracy: 97.83333539962769%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.07221689075231552 - Accuracy: 97.60000109672546%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.07360617816448212 - Accuracy: 98.04999828338623%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.93166697025299 (+- 0.19782825628334658)
> Loss: 0.069413448497653
------------------------------------------------------------------------
```

```
Test loss : 0.06353406608104706
Test accuracy: 98.07999730110168
```





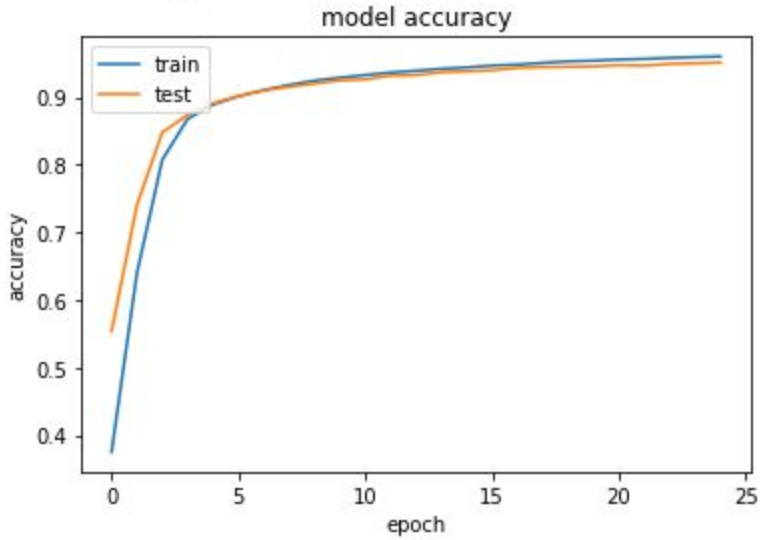
ویژگی های شبکه :

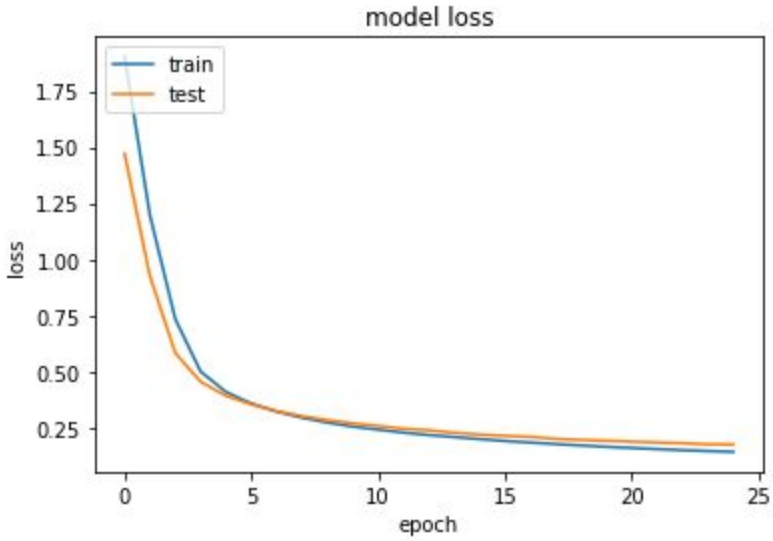
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 32 نورون و لایه دوم 16 نورون
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.18444862961769104 - Accuracy: 94.31666731834412%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.18087385594844818 - Accuracy: 94.76666450500488%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.16853468120098114 - Accuracy: 95.16666531562805%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.18647520244121552 - Accuracy: 94.51666474342346%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.1778247207403183 - Accuracy: 94.74999904632568%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.18522492051124573 - Accuracy: 94.9999988079071%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.1743919402360916 - Accuracy: 94.88333463668823%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.1940743625164032 - Accuracy: 94.45000290870667%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.18108278512954712 - Accuracy: 94.71666812896729%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.17425107955932617 - Accuracy: 95.14999985694885%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 94.77166652679443 (+- 0.2725440414613081)
> Loss: 0.1807182177901268
```

```
Test loss : 0.16390299797058105
Test accuracy: 95.24999856948853
```





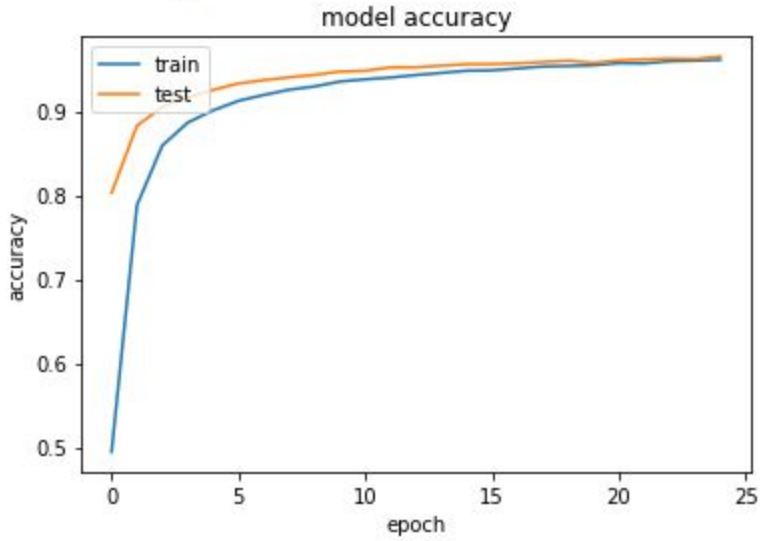
ویژگی های شبکه :

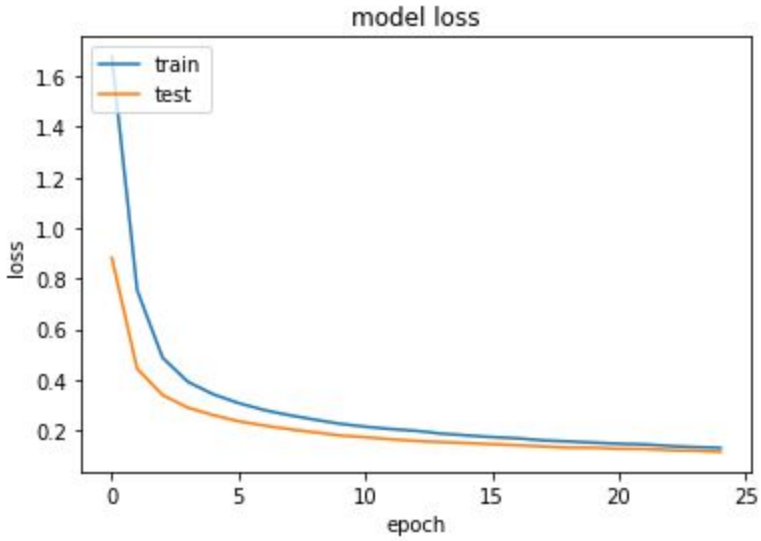
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 64 نورون و لایه دوم 32 نورون
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.10143223404884338 - Accuracy: 97.1833348274231%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.11768411099910736 - Accuracy: 96.5333342552185%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.12404879927635193 - Accuracy: 96.38333320617676%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.14233168959617615 - Accuracy: 95.95000147819519%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.1183147206902504 - Accuracy: 96.41666412353516%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.12516286969184875 - Accuracy: 95.91666460037231%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.12250251322984695 - Accuracy: 96.13333344459534%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.11555450409650803 - Accuracy: 96.61666750907898%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.1265253722667694 - Accuracy: 96.18333578109741%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.11487609893083572 - Accuracy: 96.49999737739563%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 96.38166666030884 (+- 0.35209321716218694)
> Loss: 0.12084329128265381
------------------------------------------------------------------------
```

```
Test loss : 0.10953032225370407
Test accuracy: 96.71000242233276
```





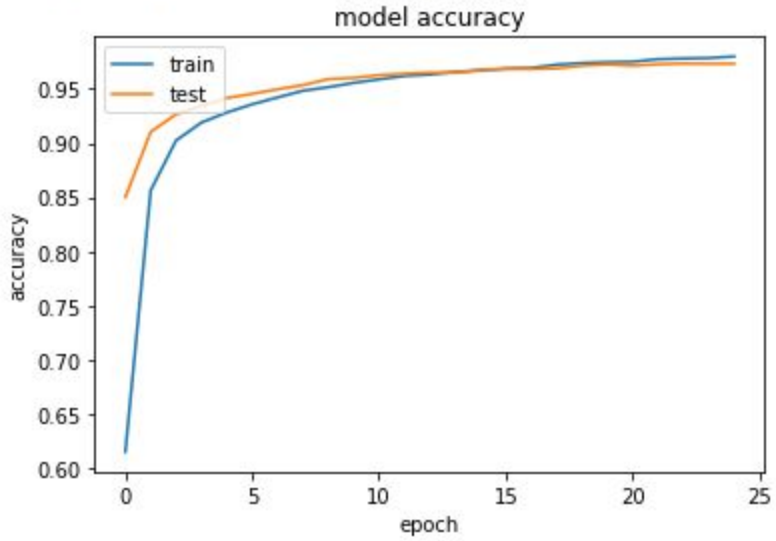
ویژگی های شبکه :

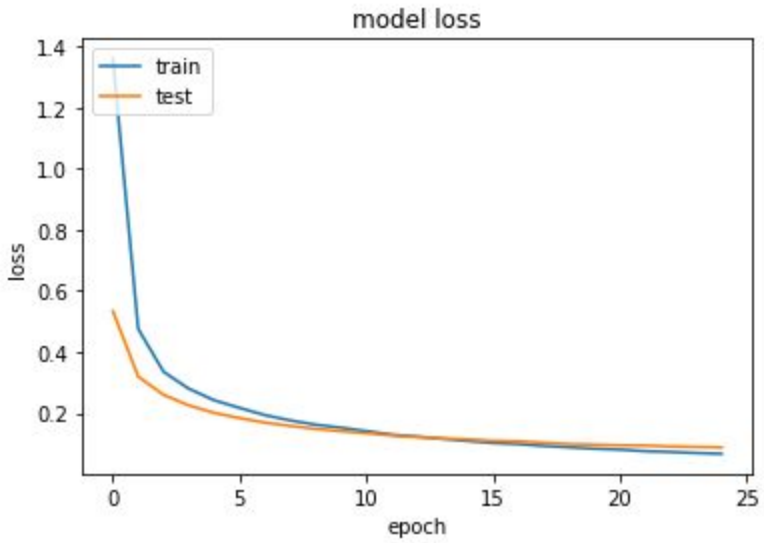
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 128 نورون و لایه دوم 64 نورون با drop out
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.08944772183895111 - Accuracy: 97.39999771118164%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.07682690769433975 - Accuracy: 97.46666550636292%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.08721111714839935 - Accuracy: 96.98333144187927%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.07549691945314407 - Accuracy: 97.68333435058594%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.09055986255407333 - Accuracy: 97.2000002861023%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.09464845806360245 - Accuracy: 97.13333249092102%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.08668002486228943 - Accuracy: 97.53333330154419%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.08404869586229324 - Accuracy: 97.35000133514404%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.08222359418869019 - Accuracy: 97.66666889190674%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.08731839060783386 - Accuracy: 97.33333587646484%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.37500011920929 (+- 0.21425005855770685)
> Loss: 0.08544616922736167
------------------------------------------------------------------------
```

```
Test loss : 0.07601005584001541
Test accuracy: 97.60000109672546
```





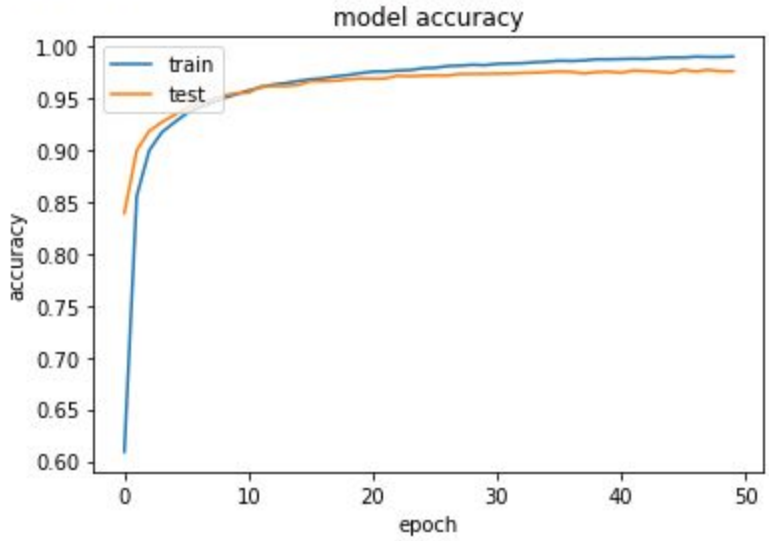
ویژگی های شبکه :

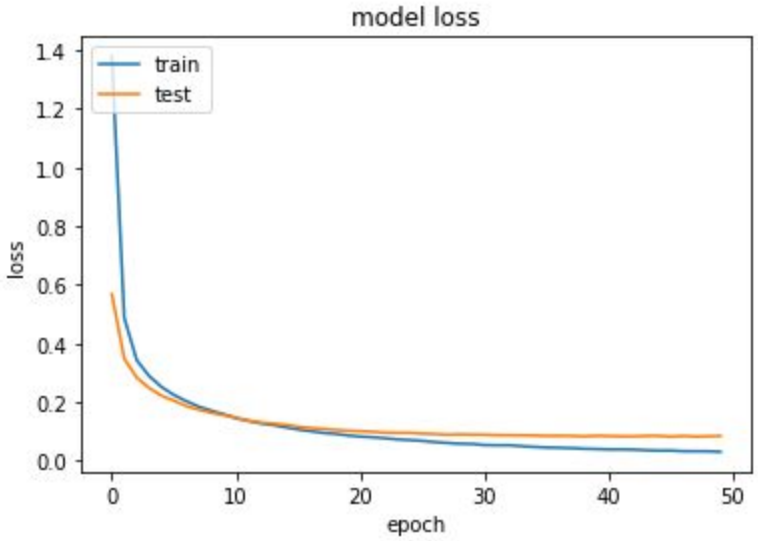
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 256 نورون و لایه دوم 128 نورون
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.06707460433244705 - Accuracy: 98.1166660785675%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.07854963839054108 - Accuracy: 98.03333282470703%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.07365447282791138 - Accuracy: 97.88333177566528%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.06874846667051315 - Accuracy: 97.93333411216736%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.065656878054142 - Accuracy: 97.91666865348816%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.0967947468161583 - Accuracy: 97.1833348274231%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.07450899481773376 - Accuracy: 98.00000190734863%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.08393444120883942 - Accuracy: 97.81666398048401%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.07354211062192917 - Accuracy: 97.83333539962769%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.08432335406541824 - Accuracy: 97.61666655540466%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.83333361148834 (+- 0.2524323370520797)
> Loss: 0.07667877078056336
------------------------------------------------------------------------
```

```
Test loss : 0.07060254365205765
Test accuracy: 97.89000153541565
```





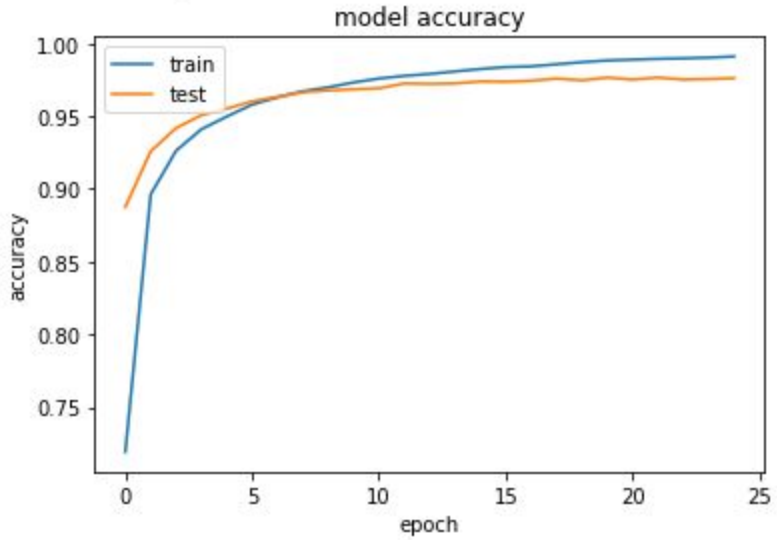
ویژگی های شبکه :

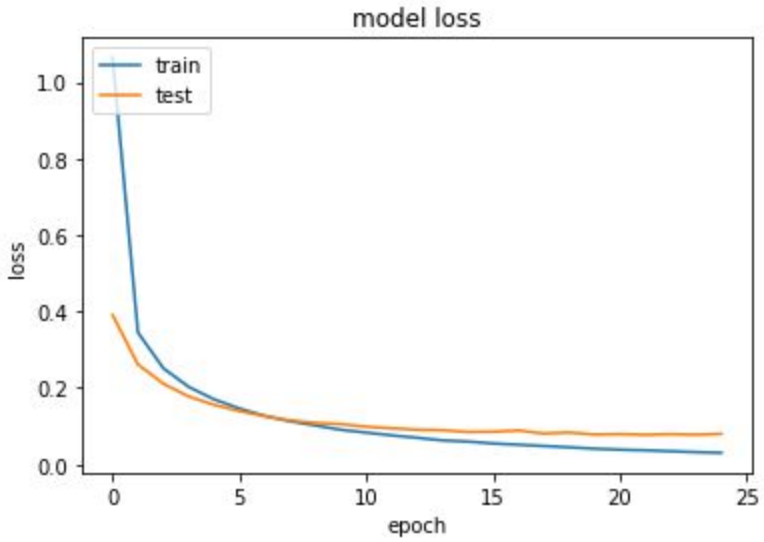
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 256 نورون و لایه دوم 128 نورون با drop out
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.06906717270612717 - Accuracy: 98.03333282470703%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.0695229098200798 - Accuracy: 97.75000214576721%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.07460198551416397 - Accuracy: 97.83333539962769%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.06075429171323776 - Accuracy: 98.0833351612091%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.0647929385304451 - Accuracy: 97.98333048820496%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.07722122222185135 - Accuracy: 97.85000085830688%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.07149498164653778 - Accuracy: 97.83333539962769%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.06615589559078217 - Accuracy: 97.86666631698608%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.0643676221370697 - Accuracy: 98.04999828338623%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.0797300636768341 - Accuracy: 97.61666655540466%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.89000034332275 (+- 0.13968161332337556)
> Loss: 0.0697709083557129
------------------------------------------------------------------------
```

```
Test loss : 0.06763458251953125
Test accuracy: 97.97000288963318
```





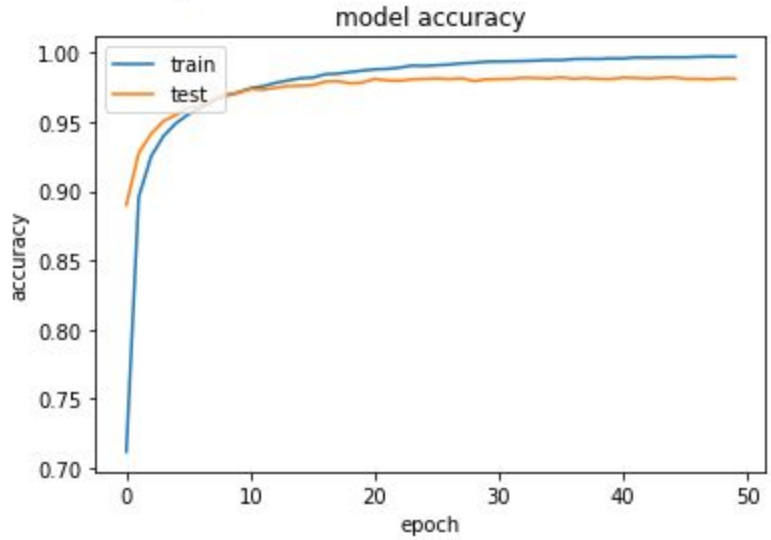
ویژگی های شبکه :

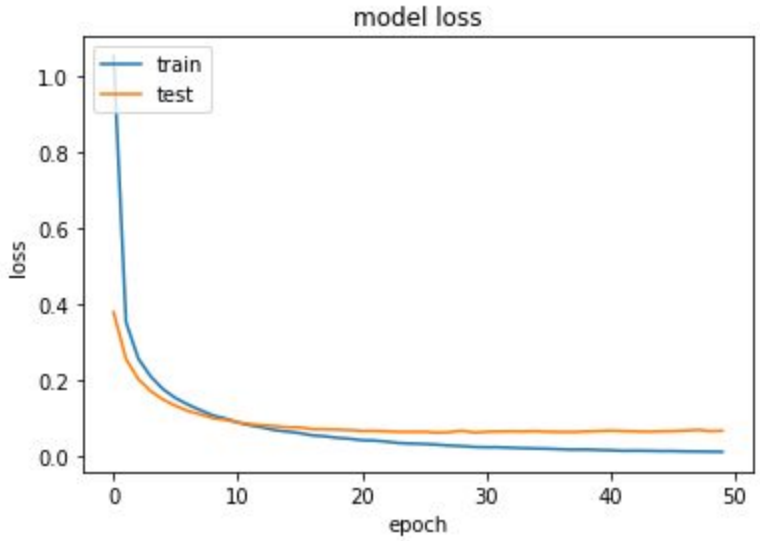
- شبکه با 60000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 256 نورون و لایه دوم 128 نورون با drop out
- Fold cross validation 10
- epoch 50

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.07261074334383011 - Accuracy: 98.0833351612091%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.06445299088954926 - Accuracy: 98.33334492279053%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.08129138499498367 - Accuracy: 97.88333177566528%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.06741143018007278 - Accuracy: 98.15000295639038%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.0730317234992981 - Accuracy: 98.1000006198883%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.06331457197666168 - Accuracy: 98.26666712760925%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.07173566520214081 - Accuracy: 98.18333387374878%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.07579946517944336 - Accuracy: 98.0833351612091%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.07232875376939774 - Accuracy: 97.89999723434448%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.06516595184803009 - Accuracy: 98.1333315372467%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 98.11166703701019 (+- 0.13376075696580247)
> Loss: 0.07071426808834076
```

```
Test loss : 0.06143460050225258
Test accuracy: 98.29999804496765
```





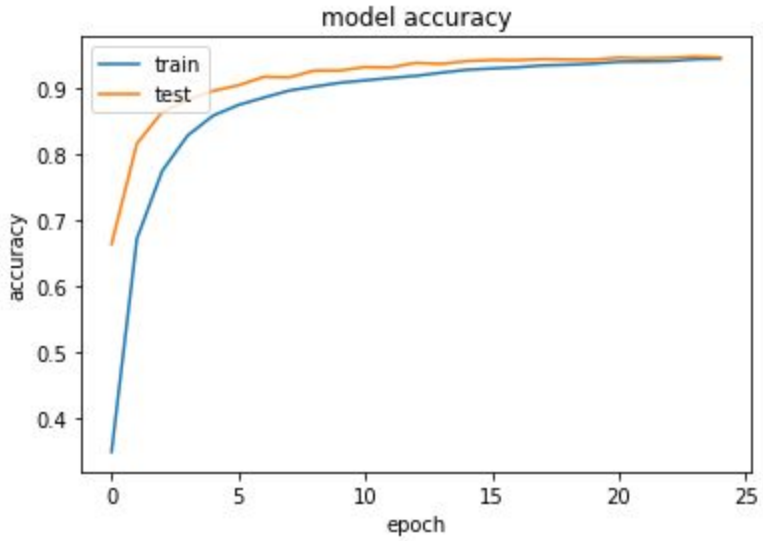
ویژگی های شبکه :

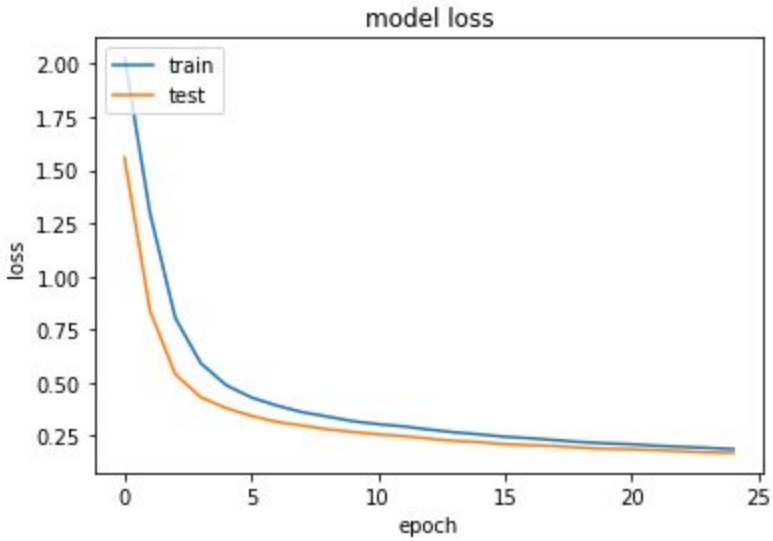
- شبکه با 30000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 256 نورون و لایه دوم 128 نورون با drop out
- Fold cross validation 10
- epoch 25

نتایج :

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.18630674481391907 - Accuracy: 94.40000057220459%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.16198281943798065 - Accuracy: 95.73333263397217%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.17416508495807648 - Accuracy: 94.70000267028809%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.1743256002664566 - Accuracy: 95.06666660308838%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.18364058434963226 - Accuracy: 94.43333148956299%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.17441426217556 - Accuracy: 94.70000267028809%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.1591058373451233 - Accuracy: 95.6333339214325%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.17281414568424225 - Accuracy: 94.30000185966492%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.173079714179039 - Accuracy: 94.9666678905487%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.16900554299354553 - Accuracy: 94.63333487510681%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 94.85666751861572 (+- 0.47165157599710034)
> Loss: 0.1728840336203575
```

```
Test loss : 0.16796761751174927
Test accuracy: 95.03999948501587
```



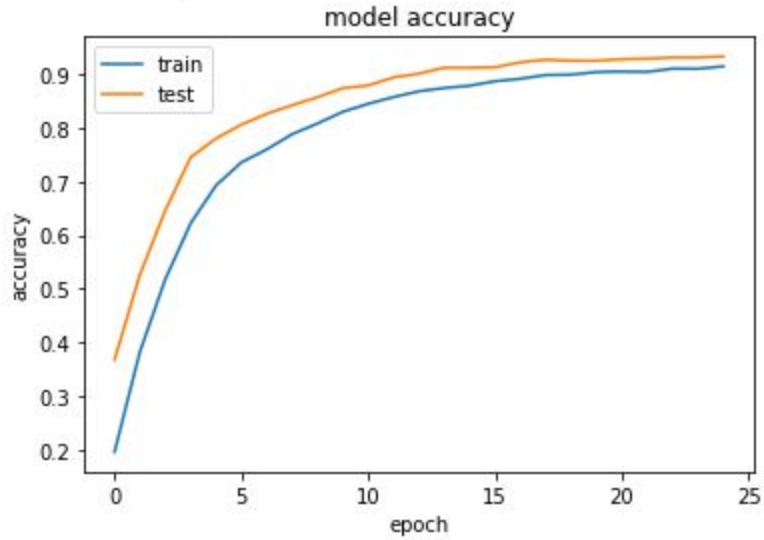


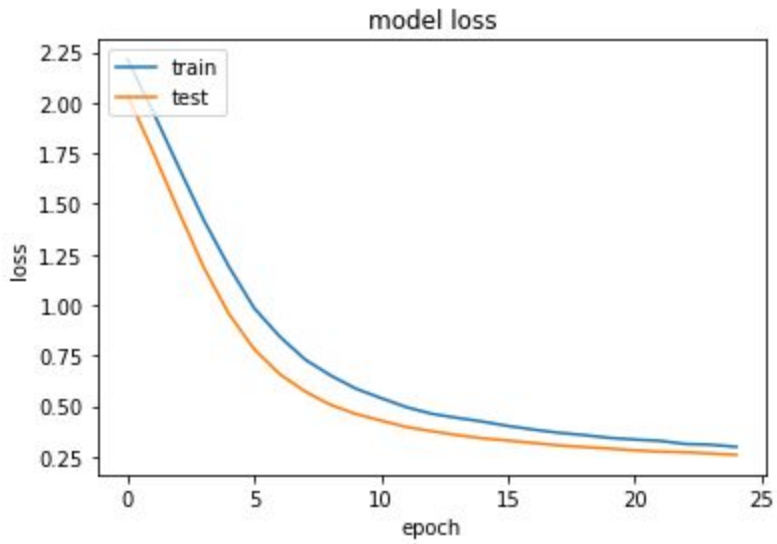
ویژگی های شبکه :

- شبکه با 10000 داده آموزشی و 10000 داده تست
- 2 لایه پنهان : لایه اول 256 نورون و لایه دوم 128 نورون با drop out
- Fold cross validation 10
- epoch 25

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.28629380464553833 - Accuracy: 91.90000295639038%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.3045411705970764 - Accuracy: 90.39999842643738%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.2805510461330414 - Accuracy: 92.00000166893005%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.2543896436691284 - Accuracy: 93.09999942779541%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.2324485033750534 - Accuracy: 93.90000104904175%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.2895372807979584 - Accuracy: 91.60000085830688%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.3216446042060852 - Accuracy: 91.00000262260437%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.2451297789812088 - Accuracy: 94.19999718666077%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.27546268701553345 - Accuracy: 91.00000262260437%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.2592191994190216 - Accuracy: 93.30000281333923%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 92.24000096321106 (+- 1.2451500434558334)
> Loss: 0.27492177188396455
------------------------------------------------------------------------
```

```
Test loss : 0.2808162271976471
Test accuracy: 92.04000234603882
```





نتایج کلی :

مشاهده می شود با افزایش تعداد نورون ها به تعداد مناسب افزایش خوبی در دقت مدل داریم. بعد از آن با افزایش بی رویه تعداد نورون ها با افزایش دقت در داده آموزشی و کاهش دقت در داده تست مواجه هستیم که نشان دهنده overfit کردن بر روی داده آموزشی و به بیان ساده تر حفظ کردن آن توسط مدل است.

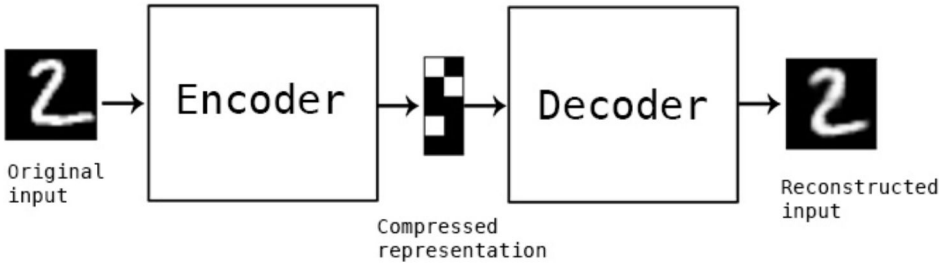
همچنین افزایش تعداد لایه ها به تنهایی کافی نیست و باید تعداد نورون مناسب در هر لایه وجود داشته باشد.

تعداد مناسب داده های آموزشی عامل بسیار مناسبی در روند آموزش دیدن می باشد. داده های کم موجب عدم آموزش صحیح می شود. داده های بیشتر موجب یادگیری بهتر می شود به شرطی که گستردگی مورد نظر را داشته باشد تا شبکه به خوبی آموزش ببیند.

بخش پنجم

https://colab.research.google.com/drive/1oZEbi6WkaITR4Txo1KqmE9IrU2qE71N1?usp=sharing

در این بخش قصد داریم از شبکه عصبی برای کاربرد رفع نویز استفاده کنیم. با تحقیقاتی که انجام دادم به این نتیجه رسیدم که برای این کار از شبکه های autoencoder استفاده می کنند که در این موارد بسیار خوب عمل می کنند. ساختار کلی این شبکه ها به صورت زیر می باشد.



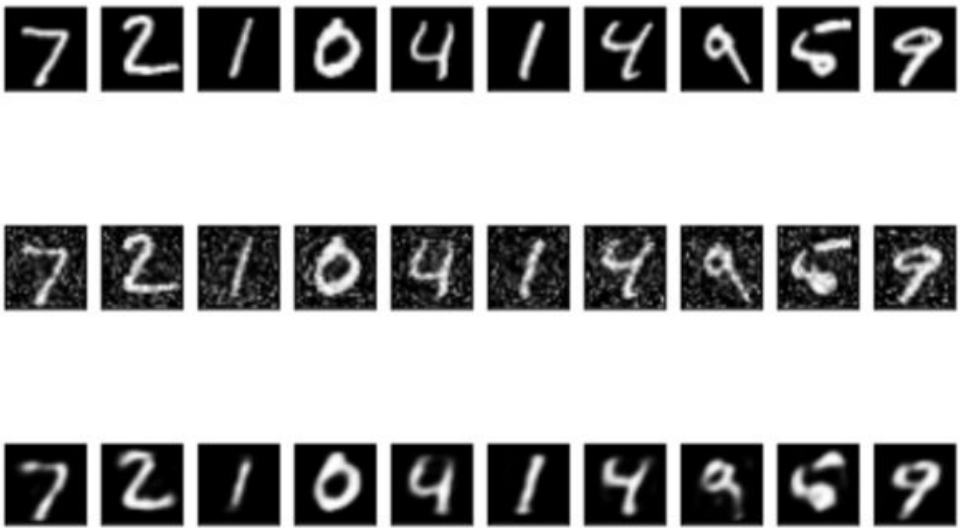
این شبکه به روش های مختلفی قابل پیاده سازی می باشد. به طور مثال به صورت fully connected , sequence to sequence Convolutional قابل پیاده سازی است. برای کاربرد رفع نویز تصویر شبکه Convolutional پیشنهاد می شود ولی در این جا به علت آشنایی بیشتر من با ساختار fully connected از این مدل استفاده کردم. ساختار شبکه ای که استفاده کردم به صورت مقابل است.

```
def model(x_train, y_train, x_test, y_test ):

    model = Sequential()

    model.add(Dense(64, activation='relu', input_shape=(784,)))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(784, activation = 'sigmoid'))

    model.summary()

    model.compile(optimizer='adam', loss='binary_crossentropy')

    history = model.fit(x_train, y_train,
            batch_size=512, epochs=25, verbose=1,
            validation_data=(x_train, y_train))

    denoise_image_test = model.predict(x_test)

    return history, denoise_image_test
```

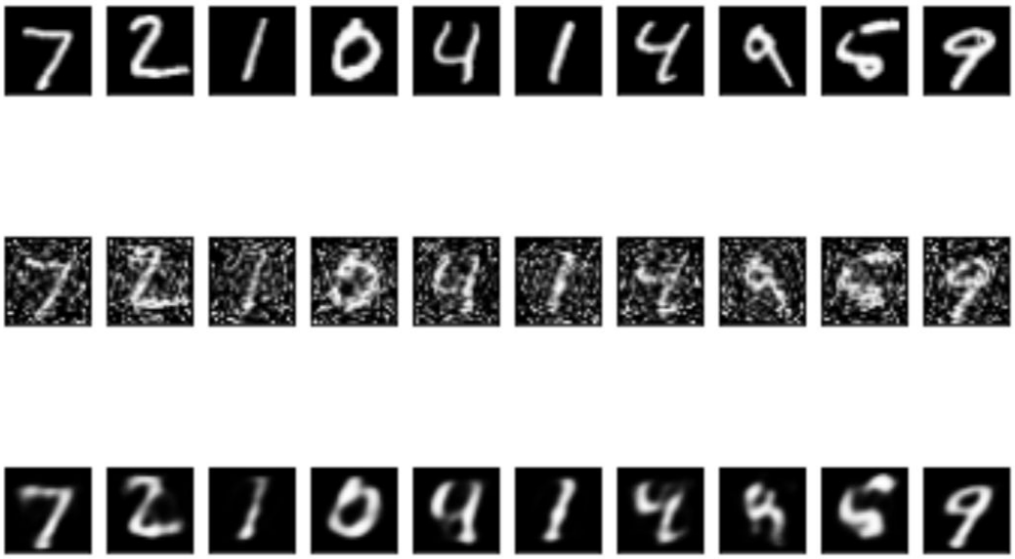
به تصویر نویز گوسی از مقدار کم تا مقدار بالا اضافه کردم. نتایج آموزش شبکه برای رفع نویز به صورت زیر می شود:

```
def add_noise(x_train, x_test, noise_factor):

    x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
    x_test_noisy = x_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape)

    x_train_noisy = np.clip(x_train_noisy, 0., 1.)
    x_test_noisy = np.clip(x_test_noisy, 0., 1.)

    return x_train_noisy, x_test_noisy
```

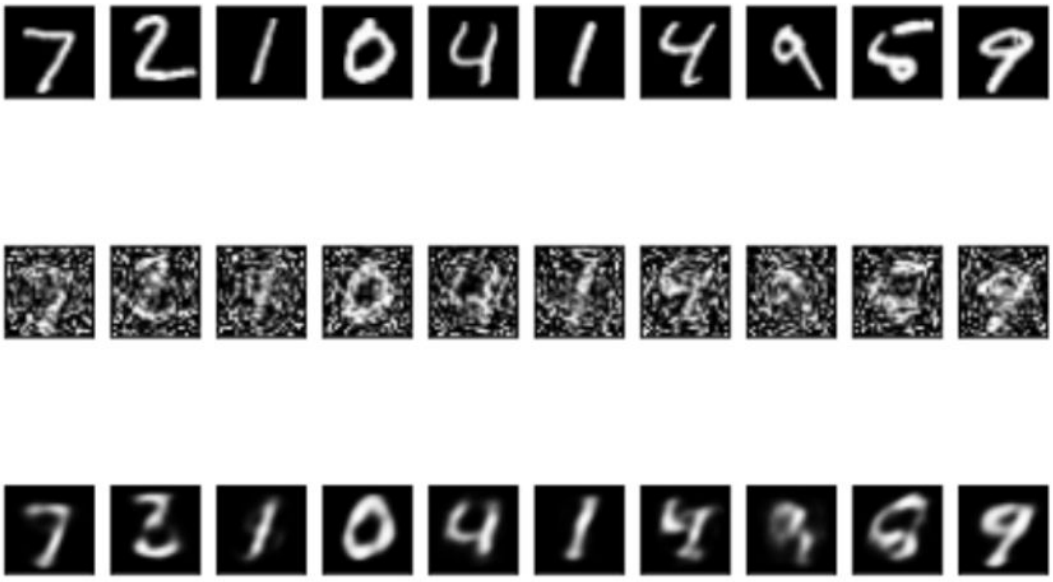
ابتدا مقدار 0.3 نویز به تصویر اضافه کردم.

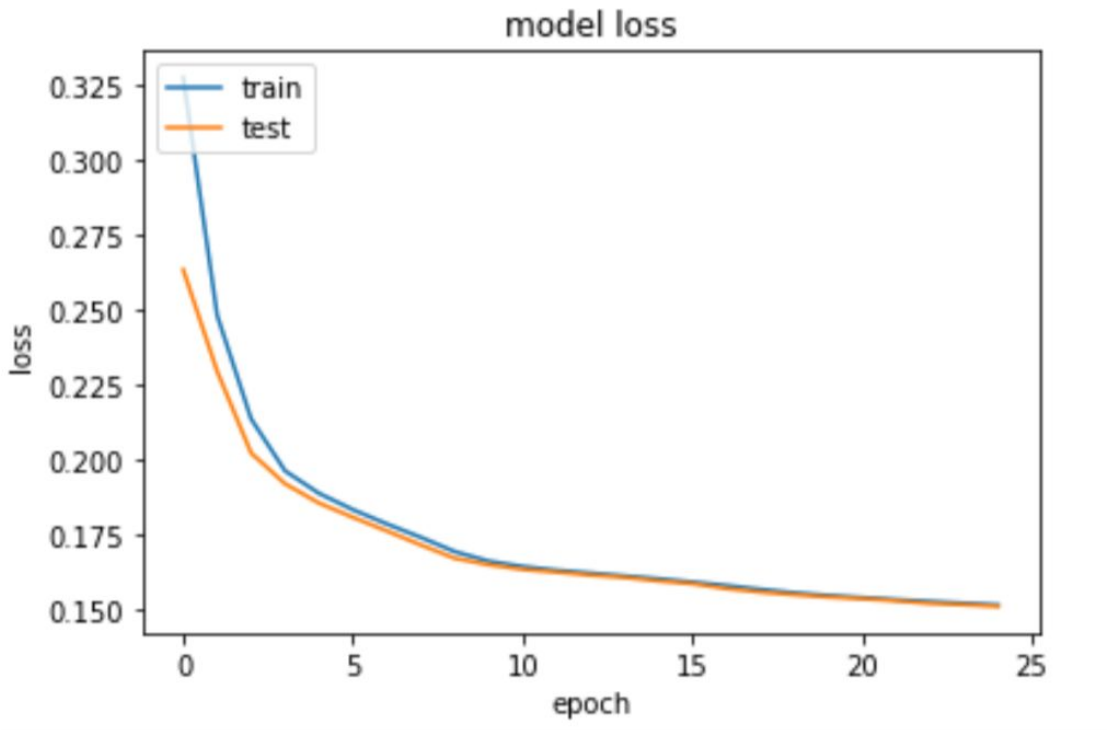


سپس 0.5 نویز به تصویر اضافه کردم.

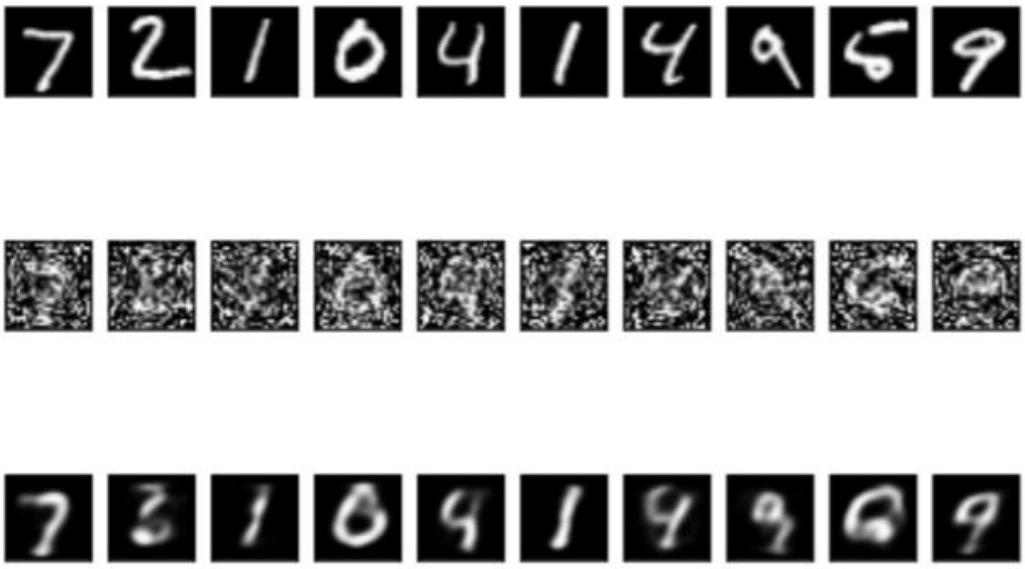


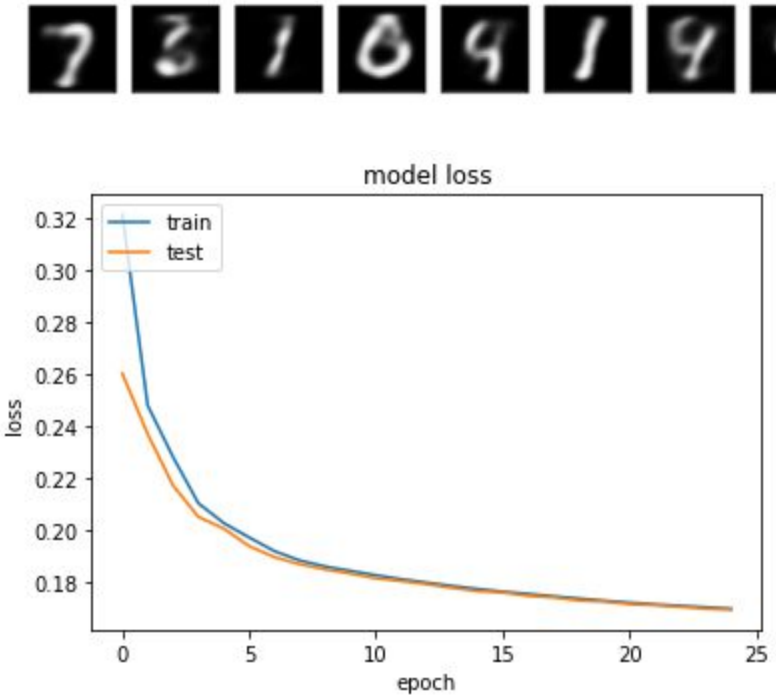
سپس 0.7 نویز به تصویر اضافه کردم.





و در نهایت 0.9 نویز به تصویر اضافه کردم.



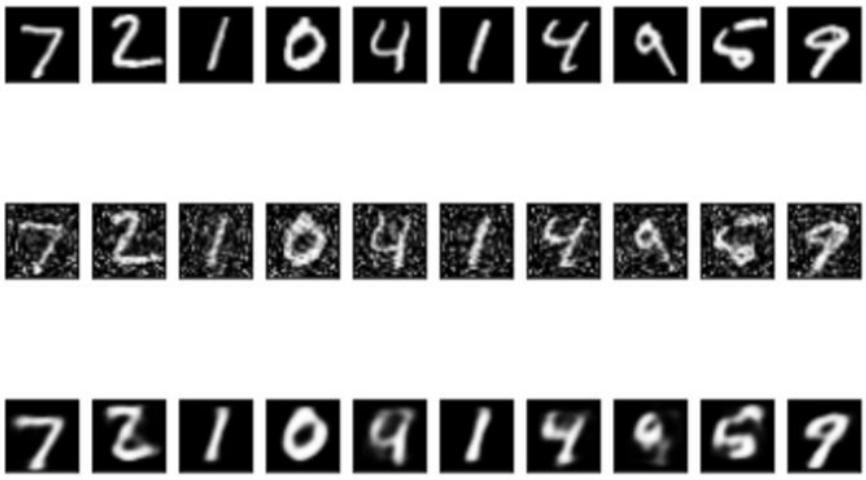


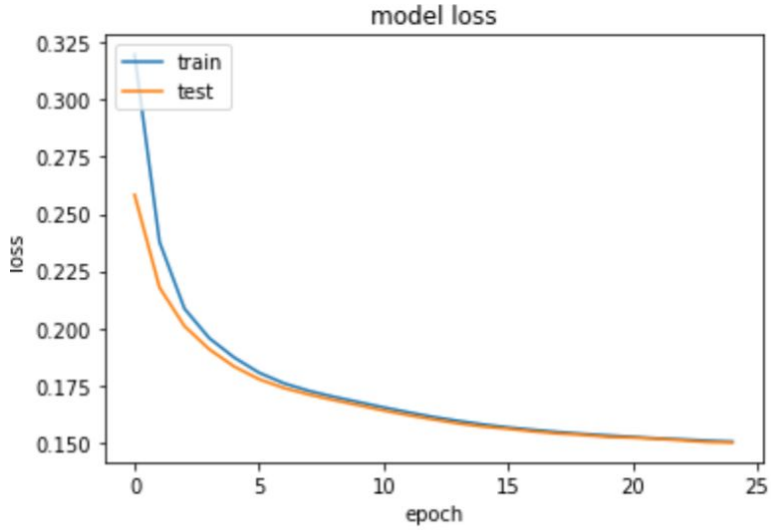
مشاهده کردم که این شبکه ساده که اندازه نه چندان بزرگی دارد در نویز های کم به خوبی آموزش می بیند و می تواند نویز را به خوبی رفع کند. در مقدار نویز بالا هم تا حد خوبی تصویر را رفع نویز میکند ولی کمی از کیفیت وضوح تصویر کاسته می شود.

نتیجه loss روی داده های آموزشی و تست تقریبا نزدیک به هم می باشد. دلیل این است که دیتا ست اندازه نسبتا خوبی دارد و همچنین نویز یکسانی به هر دو مجموعه اضافه شده است لذا تصمیم گرفتم مقدار نویز متفاوت را به دو مجموعه اضافه کنم و باز هم نتایج را بررسی کنم.

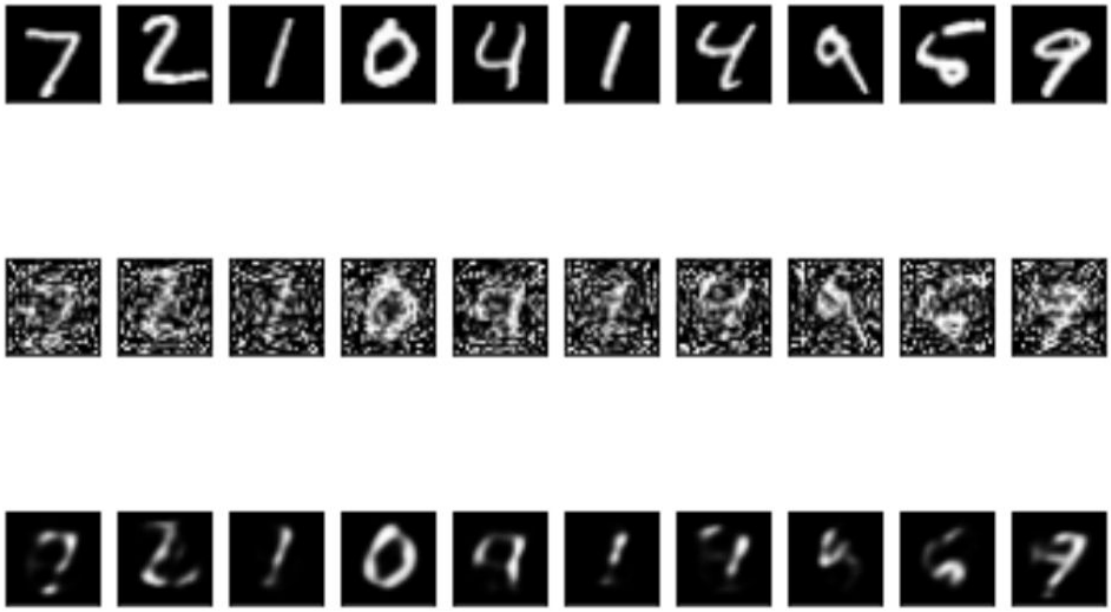
```
def add_noise(x_train, x_test, noise_factor_train, noise_factor_test):

    x_train_noisy = x_train + noise_factor_train * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
    x_test_noisy = x_test + noise_factor_test * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape)

    x_train_noisy = np.clip(x_train_noisy, 0., 1.)
    x_test_noisy = np.clip(x_test_noisy, 0., 1.)

    return x_train_noisy, x_test_noisy
```

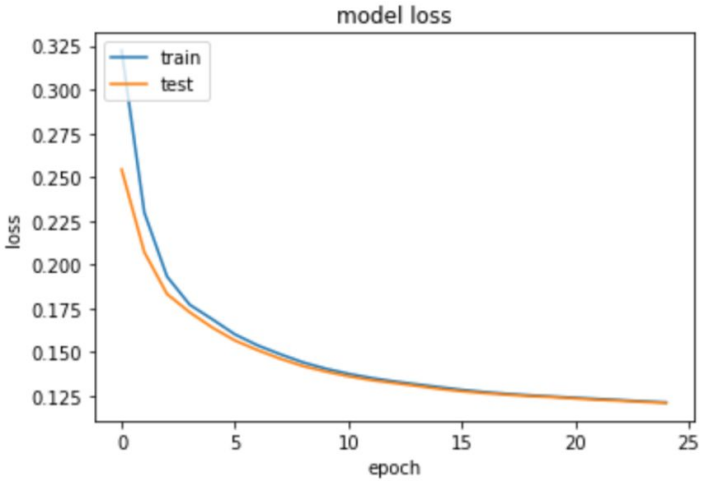
ابتدا 0.7 نویز به داده آموزشی و 0.4 به داده تست اضافه کردم.





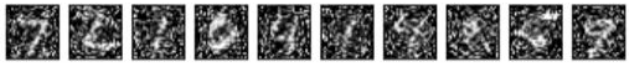
این دفعه برعکس دفعه قبل 0.4 نویز به داده آموزشی و 0.7 نویز به داده تستی اضافه کردم.

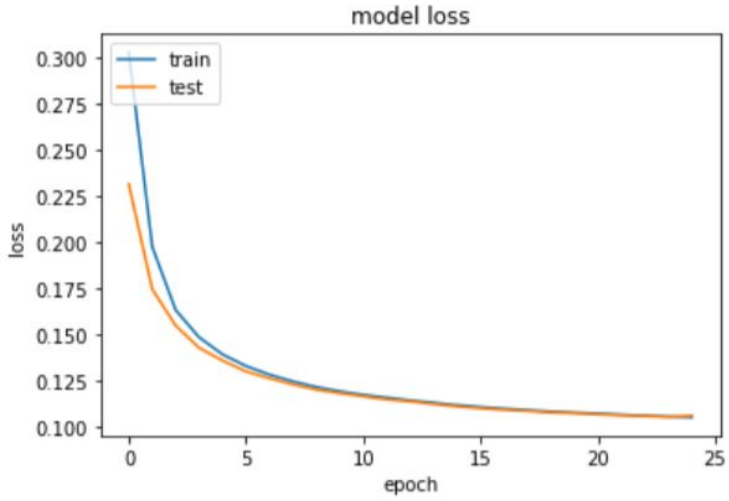




با همان میزان نویز حالت قبل این دفه شبکه را بزرگتر کردم و نتایج بهتری به دست آمد.

```
def model(x_train, y_train, x_test, y_test ):

    model = Sequential()

    model.add(Dense(128, activation='relu', input_shape=(784,)))
    model.add(Dense(64, activation='relu'))
    model.add(Dense(784, activation = 'sigmoid'))

    model.summary()

    model.compile(optimizer='adam', loss='binary_crossentropy')

    history = model.fit(x_train, y_train,
            batch_size=512, epochs=25, verbose=0,
            validation_data=(x_train, y_train))

    denoise_image_test = model.predict(x_test)

    return history, denoise_image_test
```





نتایج :

- یک شبکه fully connected با اندازه متوسط ، وقتی توسط تعداد مناسب دیتا با تعداد epoch مناسب آموزش می بیند قابلیت خوبی در رفع نویز پیدا میکند.
- بدیهی است هر چه نویز کمتر باشد شبکه بهتر می تواند آن را رفع کند.
- اگر میزان نویز داده گان تست و آموزشی به یک میزان باشد یا نویز داده های تست کمتر باشد کیفیت عکس رفع نویز شده بهتر است.
- بدیهی است با بزرگ تر کردن شبکه و افزایش تعداد epoch ها قابلیت شبکه برای رفع نویز افزایش می یابد و همچنین زمان آموزش هم افزایش می یابد و بسته به کاربرد باید بهینه ترین حالت را در نظر گرفت.
- مورد دیگر اینکه گاهی نویز خاصی در عکس وجود دارد. مثلا نویز سینوسی یا .. . بهتر است داده های آموزشی انواع نویز را داشته باشد تا مدل بهتر آموزش ببیند.

- نتیجه loss روی داده های آموزشی و تست تقریبا نزدیک به هم می باشد. دلیل این است که دیتا ست اندازه نسبتا خوبی دارد و همچنین نویز یکسانی به هر دو مجموعه اضافه شده است.