به نام خدا

دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
شبکه های عصبی و یادگیری عمیق

تمرین سری دوم

نام و نام خانوادگی	فاطمه حقیقی
شماره دانشجویی	۸۱۰۱۹۵۳۸۵
تاریخ ارسال گزارش	۱۵ فروردین ۱۳۹۹

فهرست گزارش سوالات

سوال 1 – Madaline

برای حل این سوال ابتدا ۵۰ نقطه ی تصادفی نارنجی و سبز را ایجاد کردیم. پس از آن ساختار نورون را به صورت زیر تعریف کردیم:

```
class Neuron:
  def __init__(self, weights, bias, epoch, activation_function, learning_rate):
    self.weights = weights
    self.bias = bias
    self.epoch = epoch
    self.activation_function = activation_function
    self.learning_rate = learning_rate
    self.counter = 0

  def calculate_net_value(self, inputs):
    return np.dot(inputs, self.weights) + self.bias

  def calculate_h_value(self, inputs):
    return(self.activation_function(self.calculate_net_value(inputs)))

  def update(self, target, inputs):
    self.bias = self.bias + self.learning_rate * (target - self.calculate_net_value(inputs))
    self.weights = self.weights +  np.dot(self.learning_rate * int(target - self.calculate_net_value(inputs)) , inputs)
    self.counter += 1


  def check_end_condition(self):
    if self.counter >= self.epoch:
      return True
    else:
      return False

  def get_weights(self):
    return self.weights

  def get_bias(self):
    return self.bias

  def set_weight(self, w):
    self.weights = w

  def set_bias(self, b):
    self.bias = b
```

شکل ۱: پیاده سازی ساختار نورون به زبان پایتون

پس از آن یک شبکه ایجاد کردیم که با گرفتن ورودی و تعداد نورون های لایه ی میانی، یک ساختار madaline را ایجاد می کند.

در این ساختار مقدار اولیه وزن ها و مقادیر بایاس به طور رندم و با مقادیر کوچک مقدار دهی شدند. مقادیر مناسب هایپر پارامتر در این مسئله به وسیله ی آزمون و خطا مشخص شد و به صورت زیر می باشد:

number of epochs: 200

learning rate: 0.0001

تابع مورد استفاده برای activation function در هر نورون به صورت زیر است:

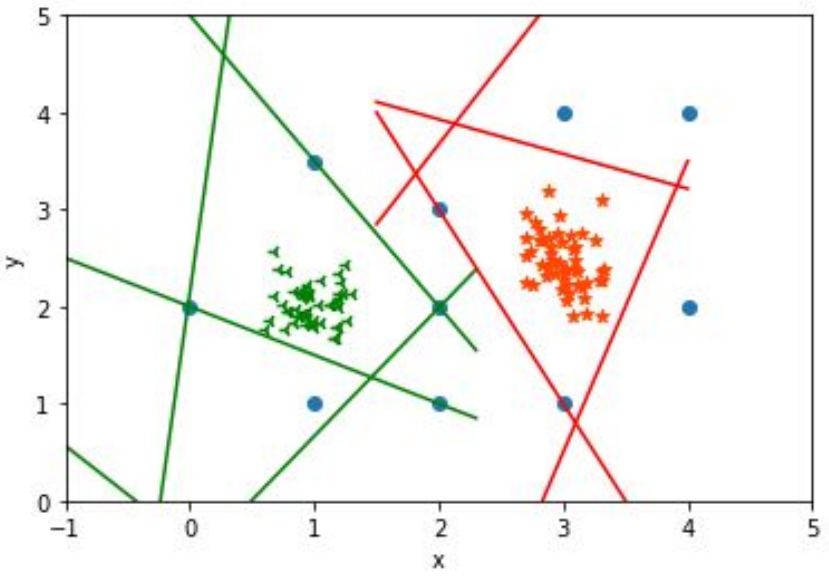
```
def activation_function(x):
  if x >= 0:
    return 1
  else:
    return -1
```

شکل ۲: پیاده سازی ساختار activation function در پایتون

پس از تعریف ساختار و مقادیر مورد نیاز برای هر دسته داده، شبکه ساخته شده را با ۱۰ نورون در لایه ی میانی آموزش دادیم زیرا با بیشتر بودن تعداد نورون، احتمال آنکه خطوط بیشتر با وزن ها ی مناسب تر پس از آموزش داشته باشیم،بیشتر است و می توان خطوط نامناسب و خارج از محدود را پس از آموزش از مجموعه خطوط حذف کرد.

پس از آموزش و تست نقاط دو مجموعه بر روی شبکه ی آموزش دیده، مقدار خطا برای نقاط سبز رنگ برابر با 0.17272727272727273 و مقدار خطا برای نقاط نارنجی رنگ برابر با 0.24545454545454545 بدست آمد.

تصویر نقاط تفکیک داده شده پس از رسم خطوط مناسب به وسیله ی وزن های آموزش دیده برای هر مجموعه داده، به صورت زیر است:



شکل ۳: تصویر نقاط تفکیک داده شده توسط چند ضلعی ایجاد شده با شبکه ی madaline

همانطور که مشاهده می شود،مجموعه نقاط سبز را به وسیله ی یک ۴ ضلعی و مجموعه نقاط نارنجی را نیز به وسیله ی یک ۴ ضلعی تفکیک کردیم.

سوال ۲ – MLP

برای این سوال از مدل sequential برای پیاده سازی شبکه استفاده کردیم.

مقادیر هایپر پارامترها در این سوال برابر است با:

number of epoch: 100

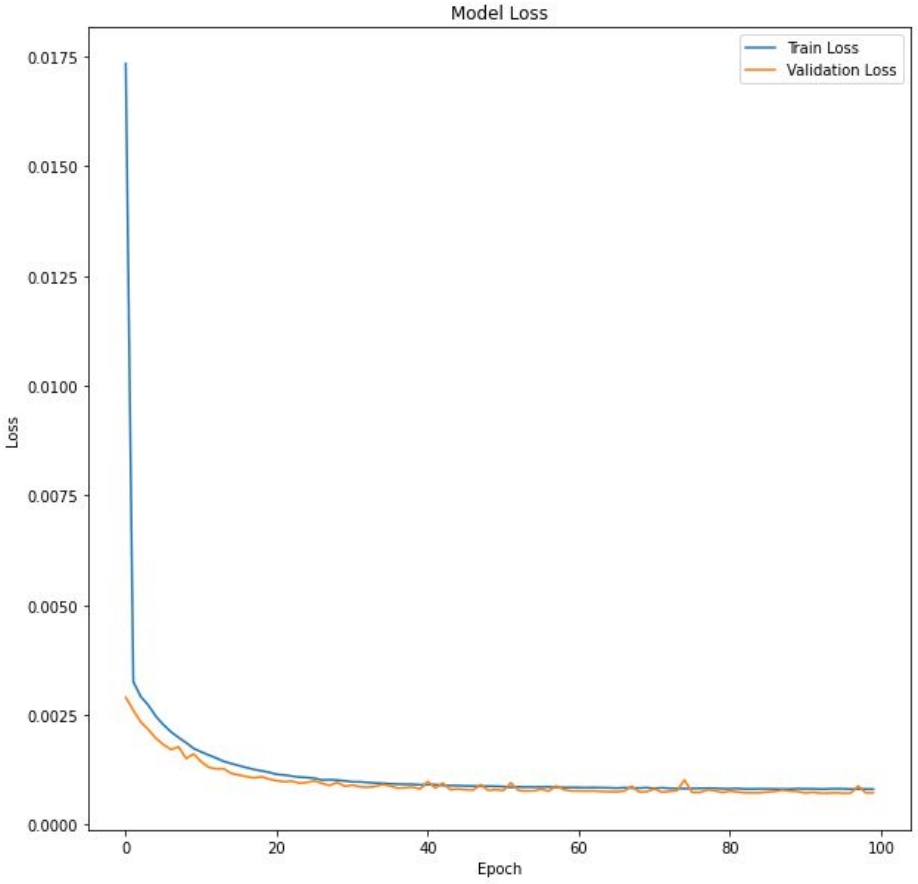
batch_size is each step = 8

learning rate: 0.01

loss function : mse

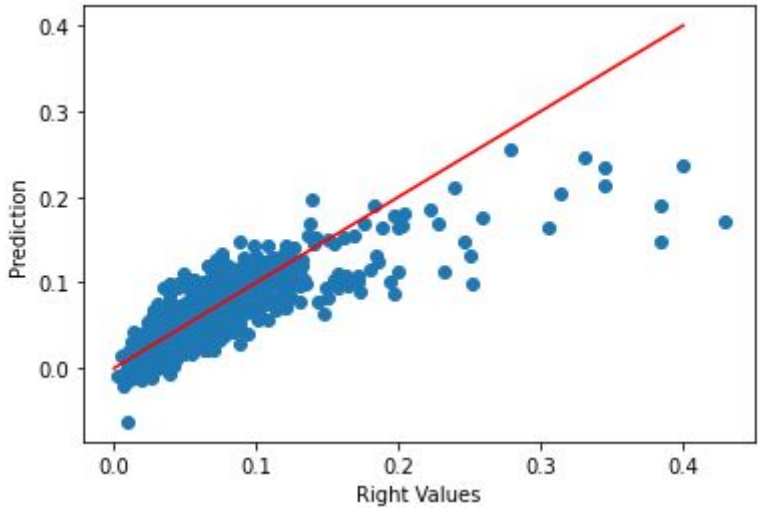
برای این سوال از مدل Stochastic mini batch based برای داده ها استفاده کردیم

در ابتدا با توجه به هایپر پارامترهای بالا یک شبکه تک لایه ساختیم، نمودار loss-epoch آن برای داده های آموزشی و داده های ارزیابی به صورت زیر است:



شکل ۴: نمودار loss-epoch برای داده های آموزشی و ارزیابی

همچنین نمودار مقادیر واقعی قیمت نسبت به مقادیر پیش بینی شده به صورت زیر است:



شکل ۵: نمودار قیمت واقعی بر حسب قیمت تخمین زده شده در شبکه با یک لایه ی مخفی

مقادیر Mean Squared Error و Mean Absoulate Erro برای داده های تست نیز به صورت زیر است:

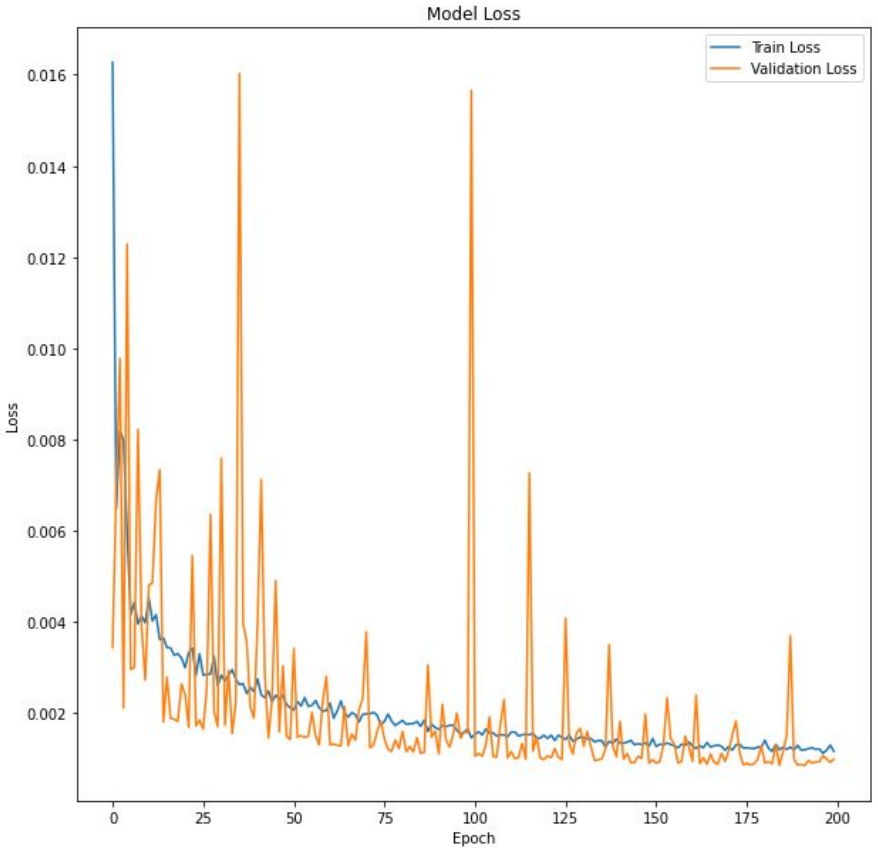
```
32/32 [==============================] - 0s 1ms/step - loss: 7.6537e-04 - mse: 7.6910e-04
Test - Mean Squared Error  0.000765365082770586
Test - Mean Absoulate Error  0.000769095728173852
```

شکل۶: تصویر MSE و MAE برای داده های تست در شبکه با یک لایه

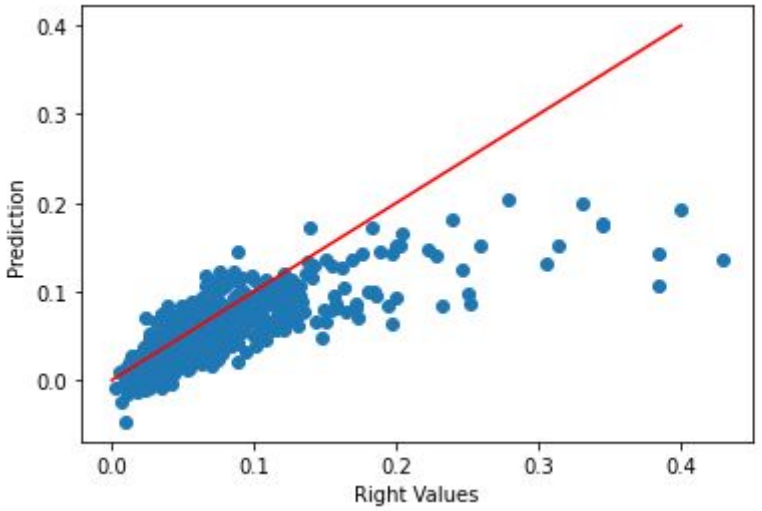
سپس یک شبکه با دو لایه مخفی ایجاد کردیم که در آن تعداد نورون های لایه ی اول ۱۵۰ و تعداد نورون های لایه ی دوم ۳۰۰ می باشد. در این شبکه تعداد epoch = 200 و

learning rate = 0.008 و loss function از نوع mse می باشد.

نمودار loss-epoch آن به صورت زیر است:



شکل ۶: نمودار loss-epoch برای شبکه با دو لایه مخفی



شکل ۷: نمودار قیمت واقعی بر حسب قیمت تخمین زده شده در شبکه با یک لایه ی مخفی

مقادیر Mean Squared Error و Mean Absoulate Erro برای داده های تست نیز به صورت زیر است:

```
Right values
32/32 [==============================] - 0s 2ms/step - loss: 0.0011 - mse: 0.0011
Test - Mean Squared Error  0.001076904241926968
Test - Mean Absoulate Error  0.0010947282426059246
```

سوال 3 – MLP

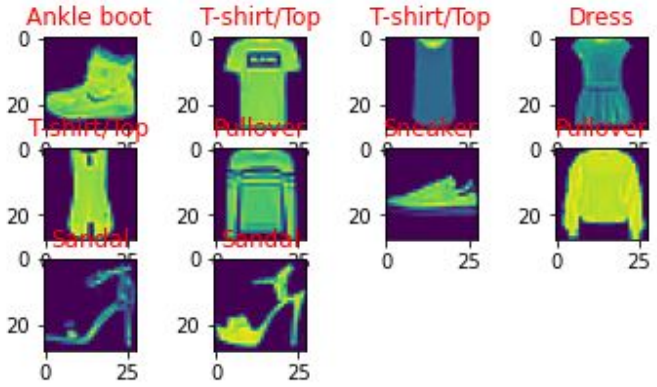
برای پیاده سازی این سوال از کتابخانه ی keras استفاده می کنیم.

ابتدا به وسیله ی کد زیر dataset fashion_mnist را دانلود می کنیم.

```
(train_img, train_label) , (test_img, test_label) = fashion_mnist.load_data()
```

همانطور که در بالا نشان داده شده، به وسیله ی keras از ۶۰ هزار تصویر اول به عنوان داده ی train و از ۱۰ هزار تصویر بعدی به عنوان داده ی تست استفاده می کنیم.

۱۰ تصویر اول این dataset همراه با نام آن ها به صورت زیر می باشد:



شکل ۸ : ۱۰ تصویر اول مجموعه داده ی fashion_mnist

در این پیاده سازی، به وسیله ی آزمون و خطا توانستیم مقدار مناسب برای هایپر پارامترهای مختلف را بدست آوریم. این مقادیر عبارت است از:

learning rate : 0.001

loss function : categorical_crossentropy

epoch number : 50

neuron number in first layer : 1100

neuron number in second layer : 900

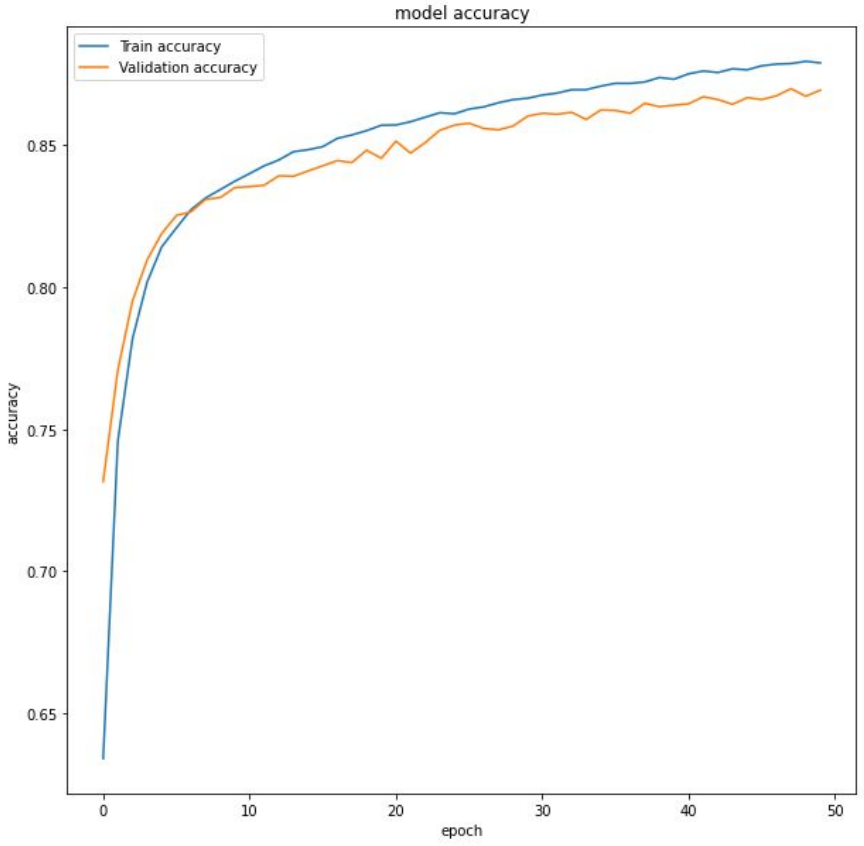
پاسخ قسمت الف)

نحوه ی تقسیم بندی داده ها به سه قسمت، train data، test data و validation date با توجه به تعداد داده ها و ساختار و عملکر شبکه صورت می گیرد. در این دسته بندی باید هر مجموعه داده، با توجه به توزیع احتمال اصلی نمونه برداری شود تا بتواند قابل استناد و نمونه ی خوبی از کل مجموعه داده باشد. در این شبکه ۸۵ درصد کل داده ها را برای train data ، و تقریبا ۱۵ درصد کل داده ها را به عنوان test data در نظر گرفتیم. همچنین ۱۰ درصد از مجموعه داده های train data را به عنوان validation data در نظر گرفتیم.

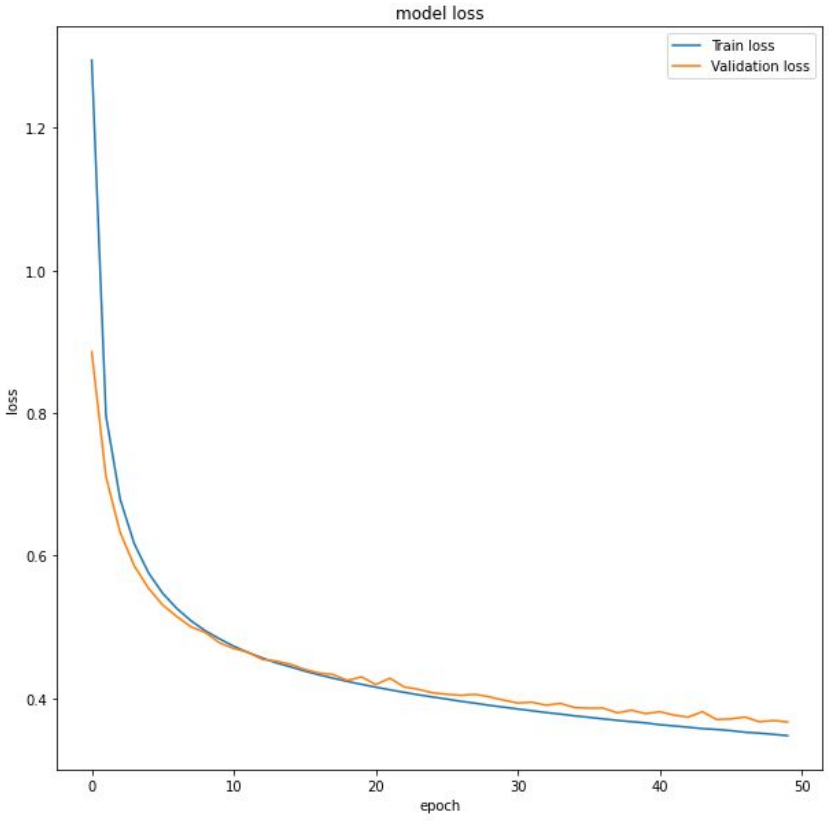
پاسخ قسمت ب)

در تمامی حالات learning rate = 0.001، تعداد epoch برابر ۵۰ ، bach_size = 32 و categorical_crossentropy : loss function می باشد.

در حالت اولیه که تعداد نورون های لایه ی اول ۱۱۰۰ و تعداد نورون های لایه ی دوم ۹۰۰ می باشد، نمودارهای accuracy-epoch و loss-epoch برای داده ی های آموزش و ارزیابی به صورت زیر می شود:



شکل ۹: نمودار accuracy-epoch برای شبکه با تعداد نورون های ۱۱۰۰ و ۹۰۰

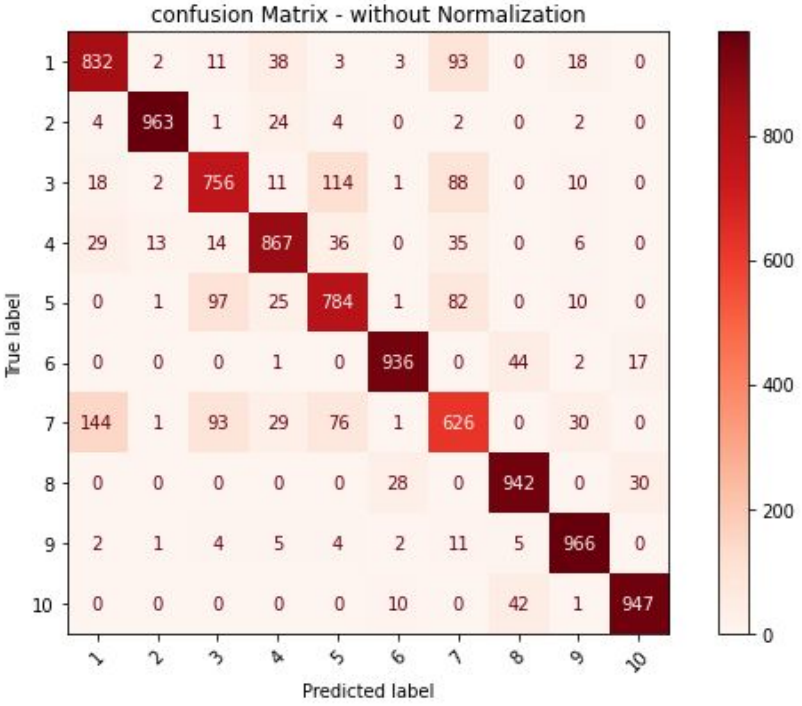


شکل ۱۰: نمودار loss-epoch برای شبکه با تعداد نورون های ۱۱۰۰ و ۹۰۰

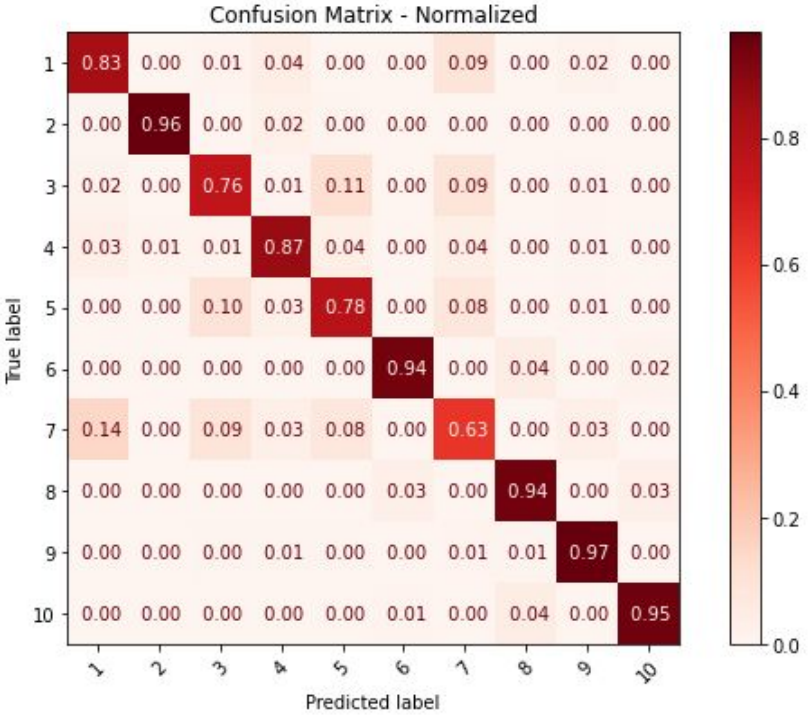
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.39408281445503235
accuracy in test data is :  0.8618999719619751
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:



شکل ۱۱: ماتریس آشفتگی بدون نرمال سازی برای شبکه با تعداد نورون های ۱۱۰۰ و ۹۰۰

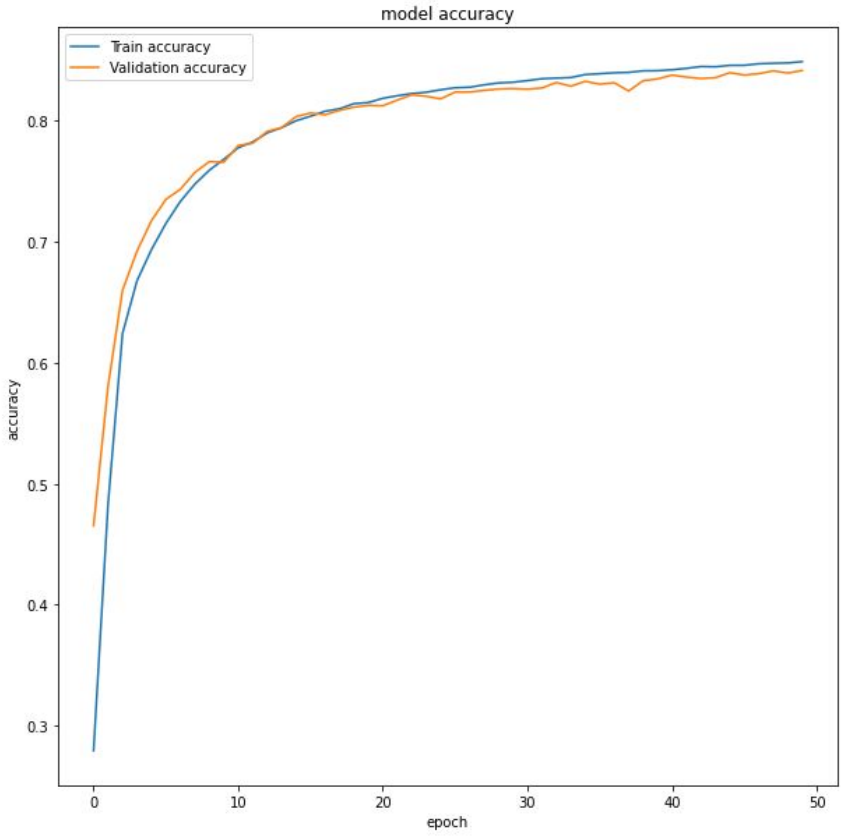


شکل ۱۲: ماتریس آشفتگی نرمال سازی شده برای شبکه با تعداد نورون های ۱۱۰۰ و ۹۰۰

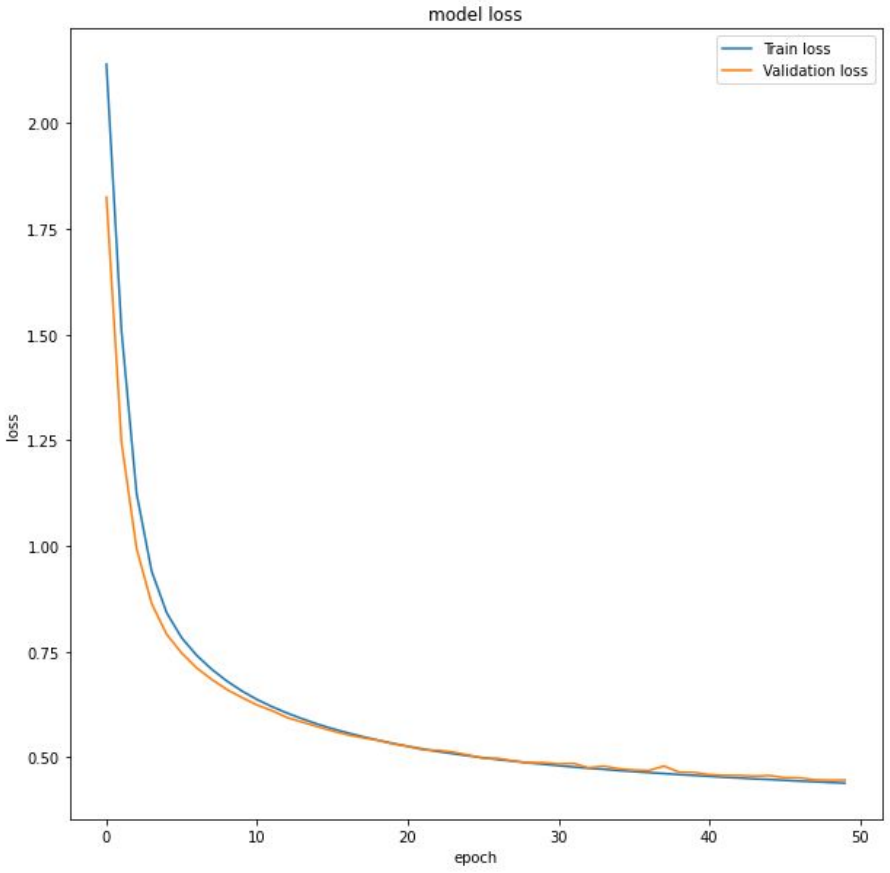
حالت اول: در صورتی که تعداد نورون های هر دو لایه را کاهش دهیم و تعداد نرورن لایه ی اول از دوم کمتر باشد:

تعداد نورون های لایه ی اول برابر با ۱۰ و تعداد نورون های لایه ی دوم برابر با ۵۴۰ باشد.

در این حالت نمودارهای accuracy-epoch و loss-epoch برای داده ی های آموزش و ارزیابی به صورت زیر می شود:



شکل ۱۳: نمودار accuracy-epoch برای شبکه با تعداد نورون های ۱۰ و ۵۴۰

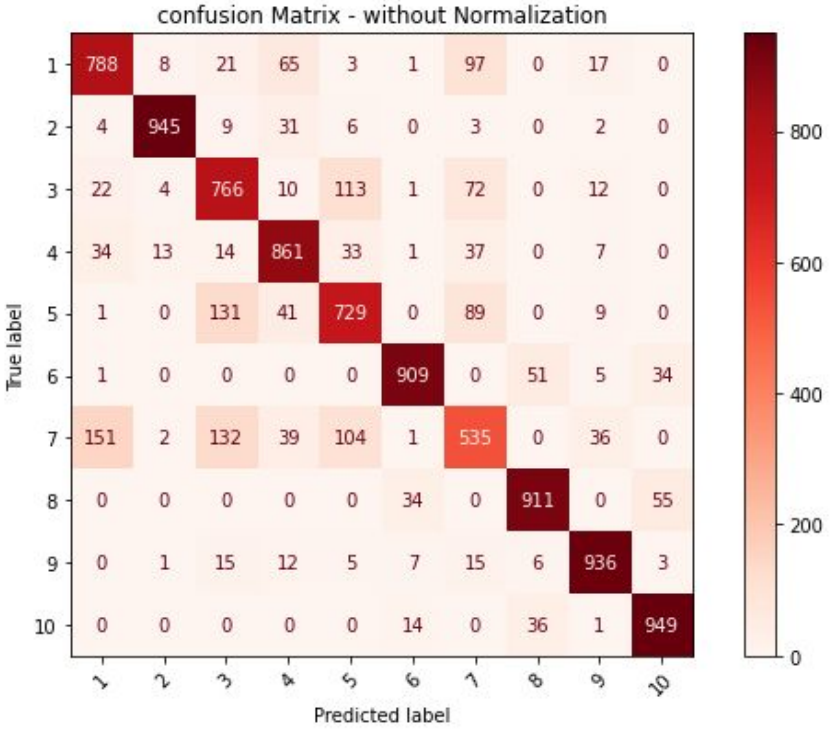


شکل ۱۴: نمودار loss-epoch برای شبکه با تعداد نورون های ۱۰ و ۵۴۰

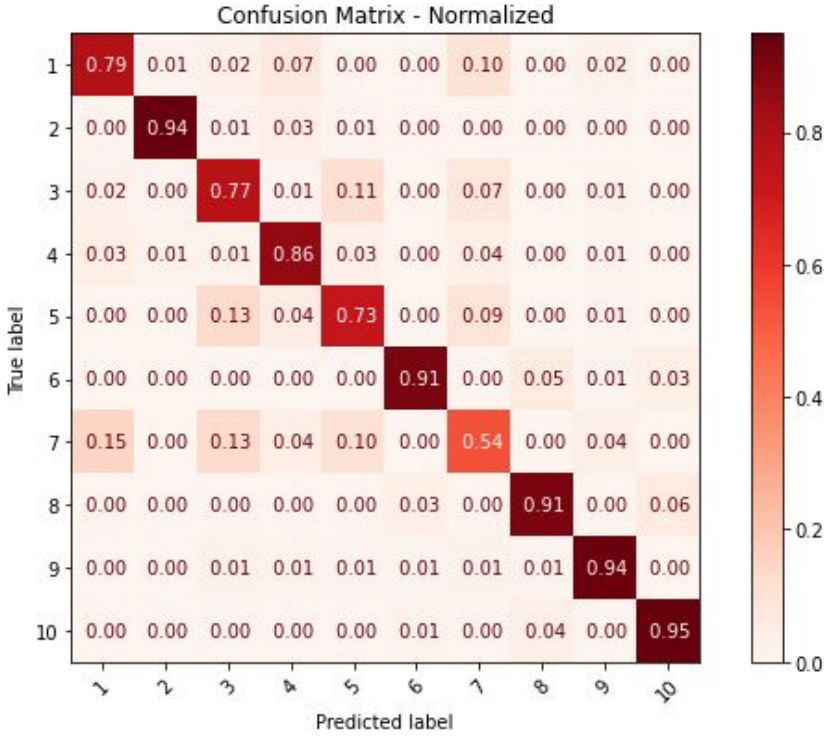
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.4766693711280823
accuracy in test data is :  0.8328999876976013
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:



شکل ۱۵: ماتریس آشفتگی نرمال سازی نشده برای شبکه با تعداد نورون های ۱۰ و ۵۴۰

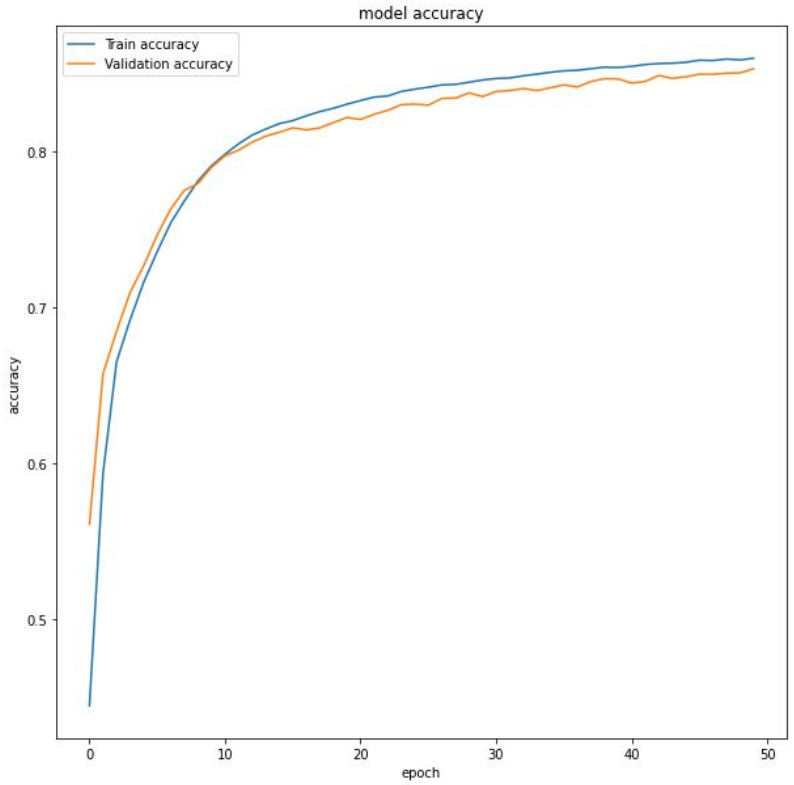


شکل ۱۶: ماتریس آشفتگی نرمال سازی شده برای شبکه با تعداد نورون های ۱۰ و ۵۴۰

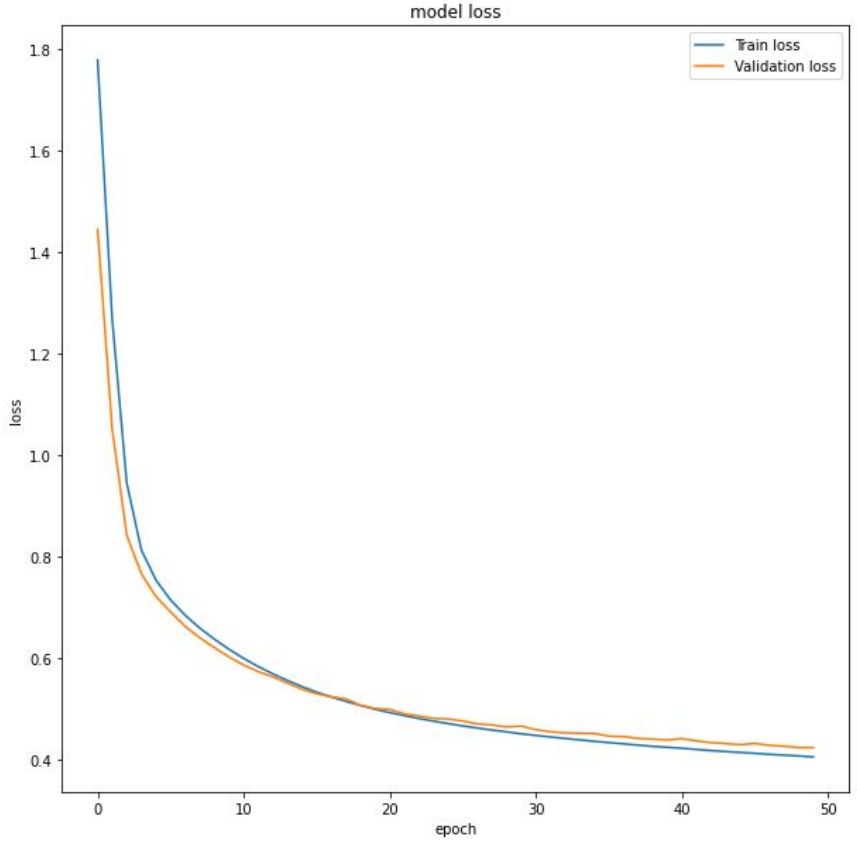
حالت دوم: حالتی که تعداد نورون های هر دو لایه کمتر باشد و تعداد نورون های لایه ی اول از لایه ی دوم بیشتر باشد:

تعداد نورون های لایه ی اول برابر با ۵۰ و تعداد نورون های لایه ی دوم برابر با ۱۰ باشد.

در این حالت نمودارهای accuracy-epoch و loss-epoch برای داده ی های آموزش و ارزیابی به صورت زیر می شود:



شکل ۱۷: نمودار accuracy-epoch برای شبکه با تعداد نورون های ۵۰ و ۱۰

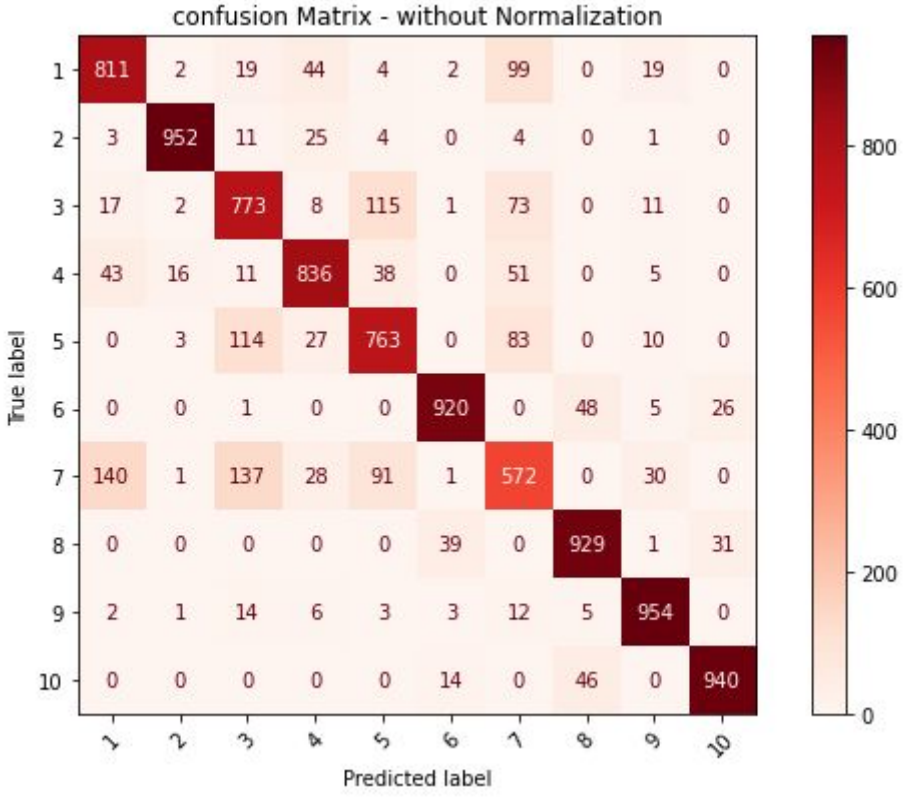


شکل ۱۸: نمودار loss-epoch برای شبکه با تعداد نورون های ۵۰ و ۱۰

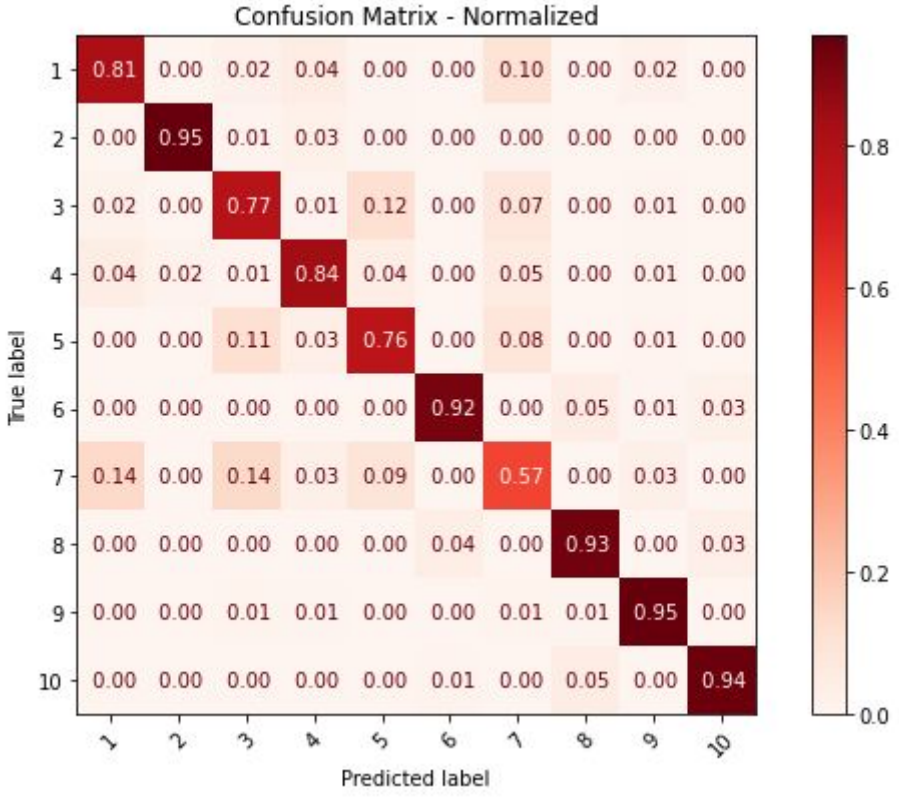
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.4476568400859833
accuracy in test data is :  0.8450000286102295
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:



شکل ۱۹: ماتریس آشفتگی نرمال سازی نشده برای شبکه با تعداد نورون های ۵۰ و ۱۰

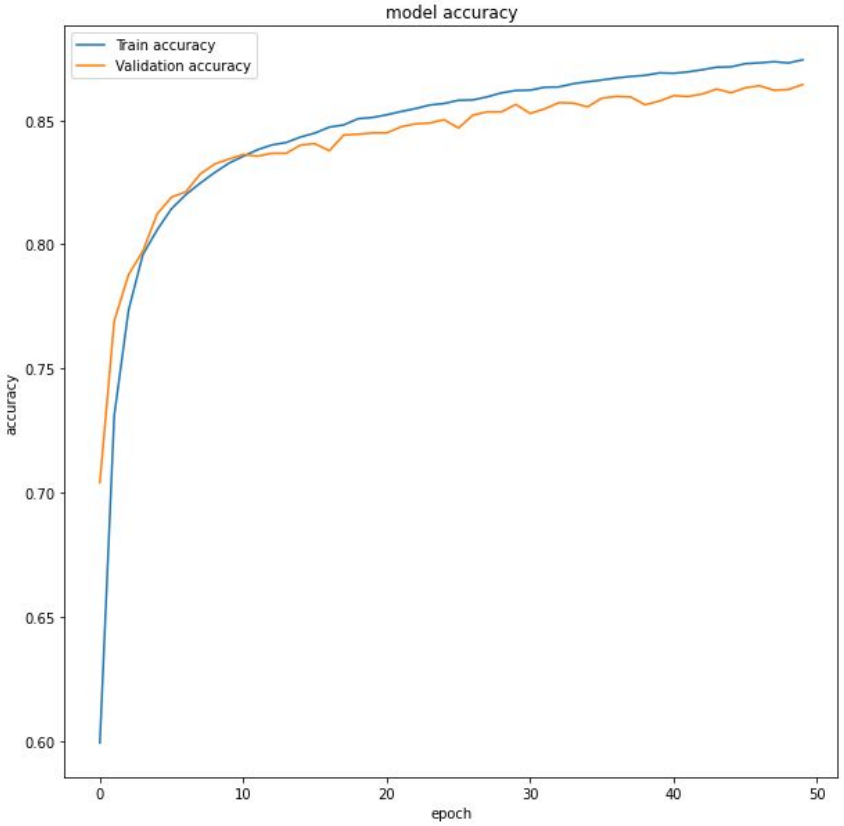


شکل ۲۰: ماتریس آشفتگی نرمال سازی شده برای شبکه با تعداد نورون های ۵۰ و ۱۰

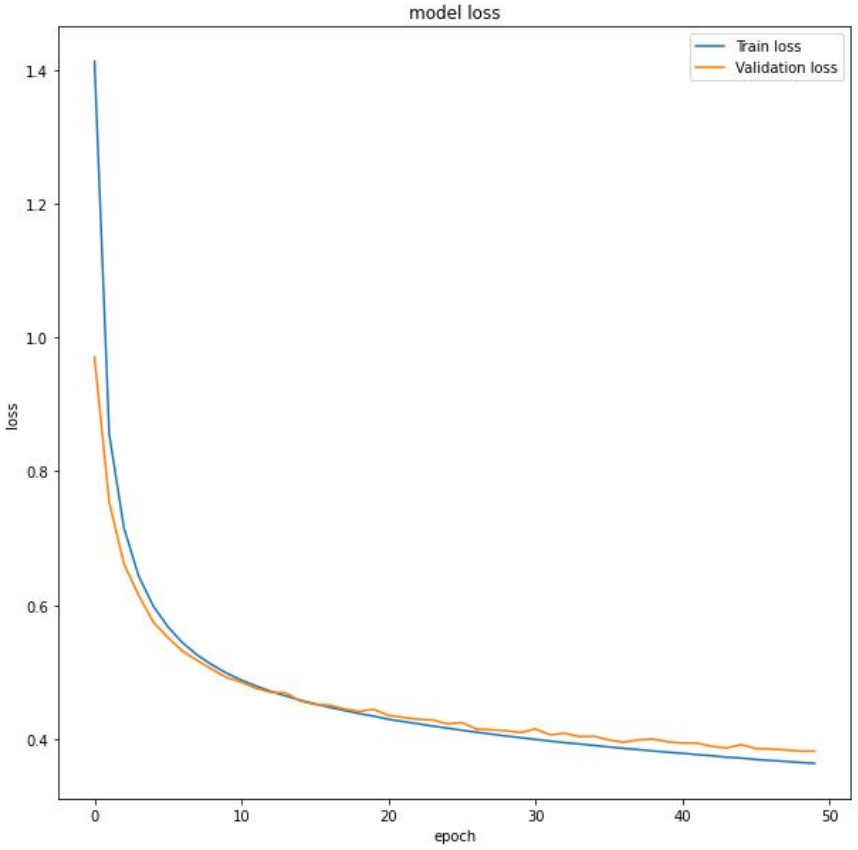
حالت سوم: حالتی که تعداد نورون های هر دو لایه کمتر باشد و تعداد نورون های لایه ی اول با لایه ی دوم برابر باشد:

تعداد نورون های لایه ی اول برابر با ۴۰۰ و تعداد نورون های لایه ی دوم برابر با ۴۰۰ باشد.

در این حالت نمودارهای accuracy-epoch و loss-epoch برای داده ی های آموزش و ارزیابی به صورت زیر می شود:



شکل ۲۱: نمودار accuracy-epoch برای شبکه با تعداد نورون های ۴۰۰ و ۴۰۰

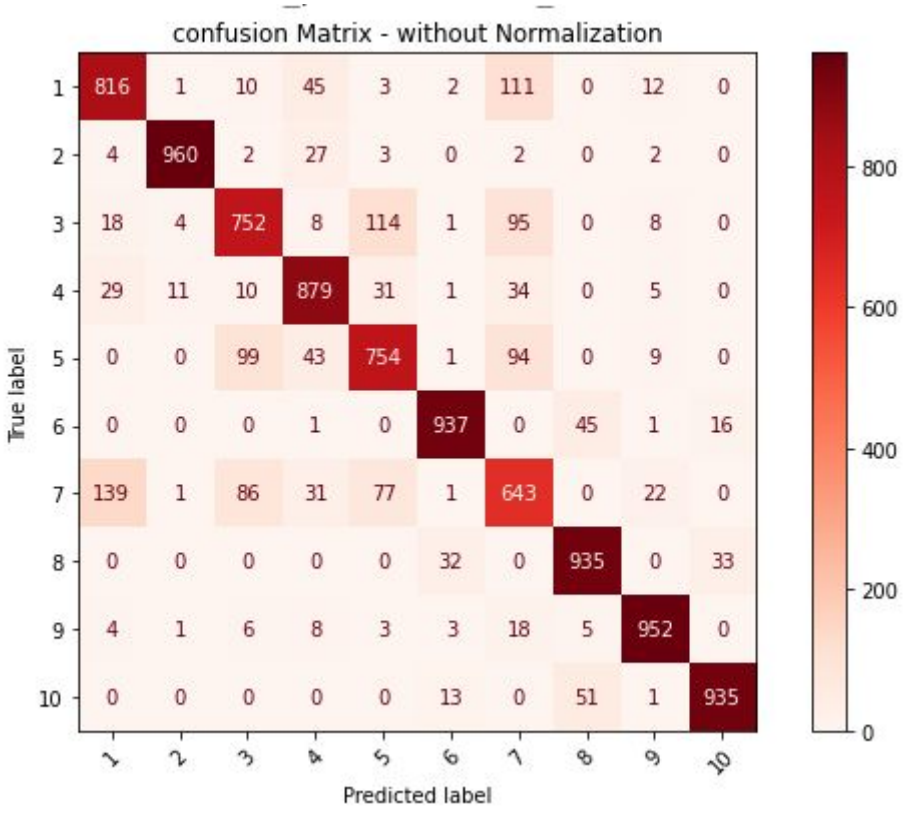


شکل ۲۲: نمودار loss-epoch برای شبکه با تعداد نورون های ۴۰۰ و ۴۰۰

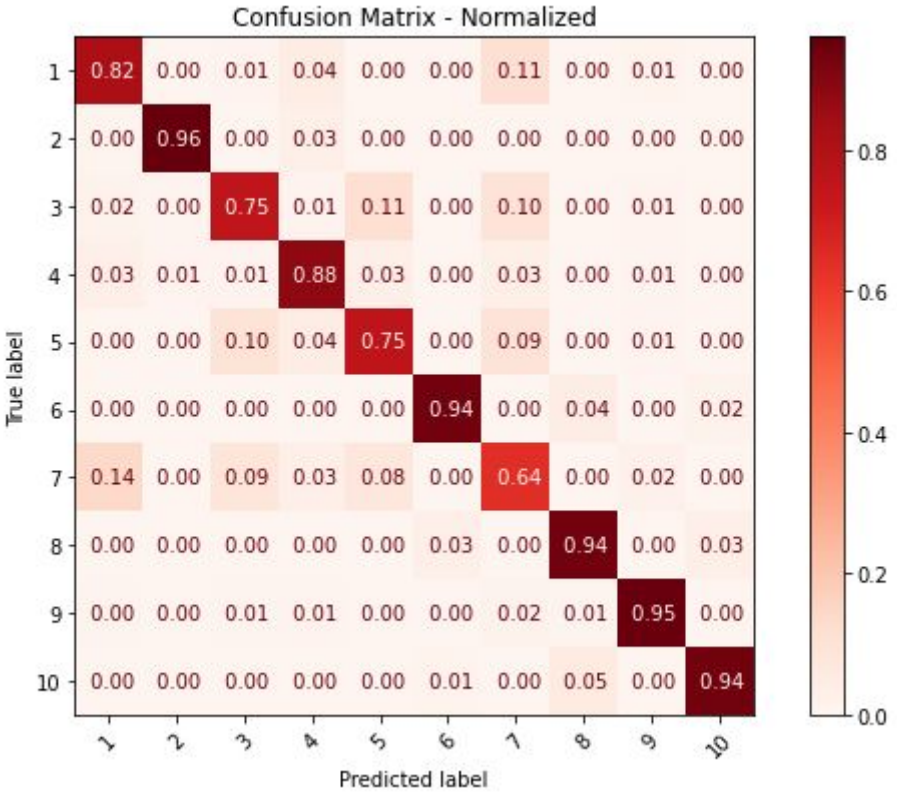
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.40598493814468384
accuracy in test data is :  0.8562999963760376
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:



شکل ۲۳: ماتریس آشفتگی نرمال سازی نشده برای شبکه با تعداد نورون های ۴۰۰ و ۴۰۰



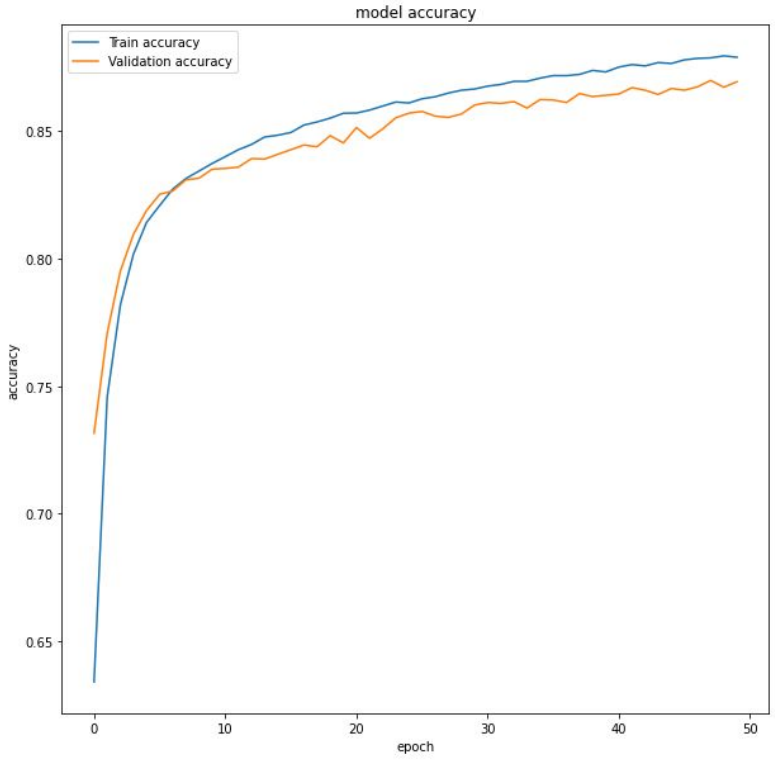
شکل ۲۴: ماتریس آشفتگی نرمال سازی شده برای شبکه با تعداد نورون های ۴۰۰ و ۴۰۰

همانطور که در بالا نشان داده شده دقت مدل برای داده ی تست در شبکه با تعداد نورون های ۴۰۰ و ۴۰۰ بیشتر از دقت مدل برای داده های تست در شبکه با تعداد نورون های ۵۰ و ۱۰ می باشد و نیز دقت مدل برای داده ی تست در شبکه با تعداد نورون های ۵۰ و ۱۰ بیشتر از دقت مدل برای داده های تست در شبکه با تعداد نورون های ۱۰ و ۵۴۰ می باشد.

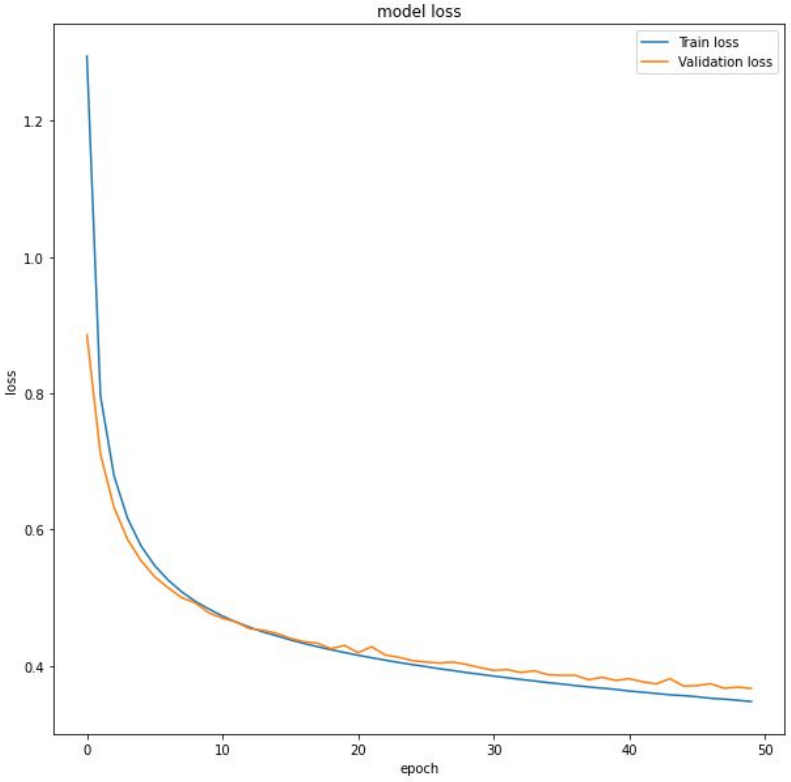
همچنین از لحاظ مدت زمان لازم برای آموزش شبکه،به ترتیب شبکه با تعداد نورون های ۱۱۰۰ و ۹۰۰ از نسبت به شبکه های دیگر زمان بیشتری برای آموزش لازم داشت و پس از آن شبکه با تعداد نورون های ۵۰ و ۱۰ زمان بیشتری نسبت به دو شبکه ی دیگر ، سپس شبکه با تعداد نورون های ۴۰۰ و ۴۰۰ زمان بیشتری نسبت به شبکه با تعداد نورون های ۱۰ و ۵۴۰ لازم داشت

پاسخ قسمت ج)

در حالت batch size = 32 نمودار های accuracy-epoch و loss-epochبه صورت زیر می باشد:



شکل ۲۵: نمودار accuracy-epoch برای شبکه با batched_size = 32

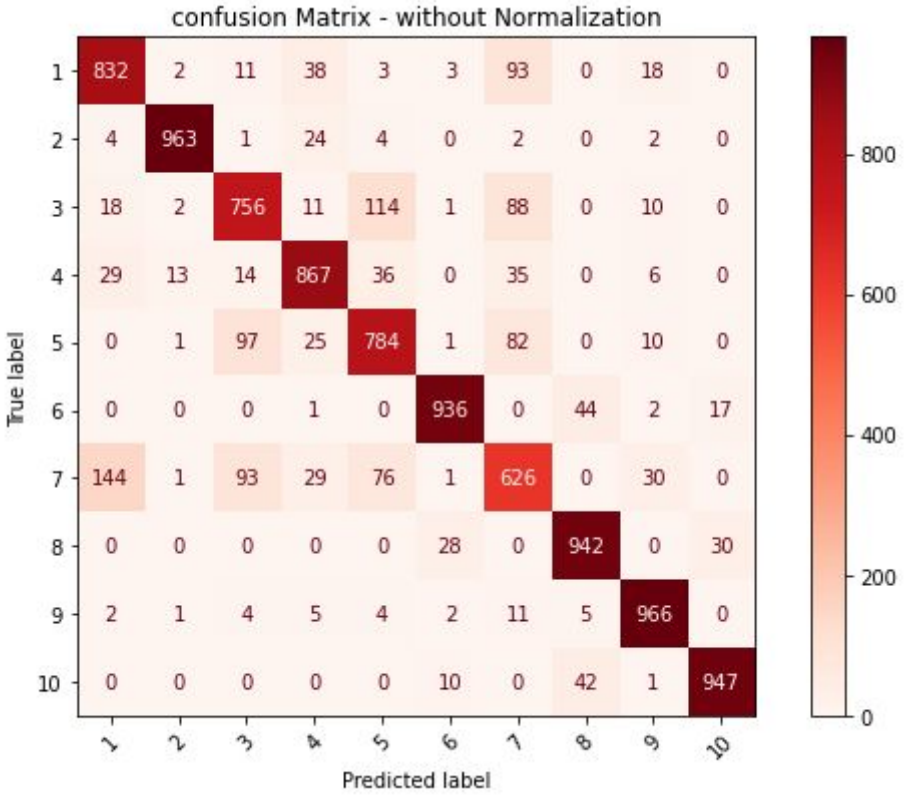


شکل ۲۶: نمودار loss-epoch برای شبکه با batched_size = 32

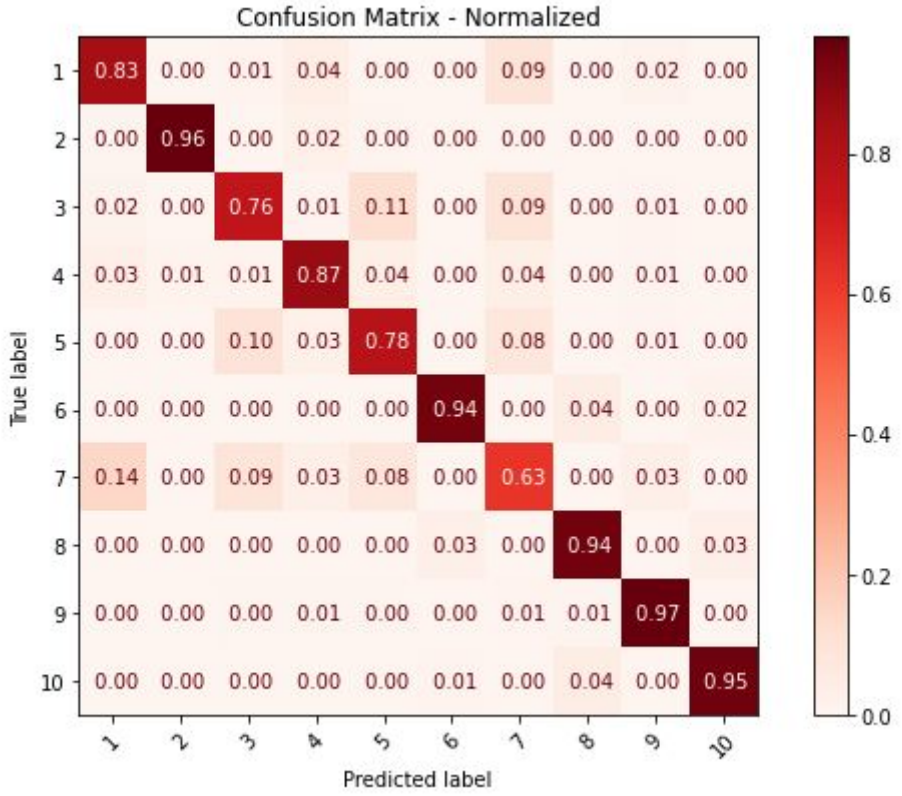
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.39408281445503235
accuracy in test data is :  0.8618999719619751
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:

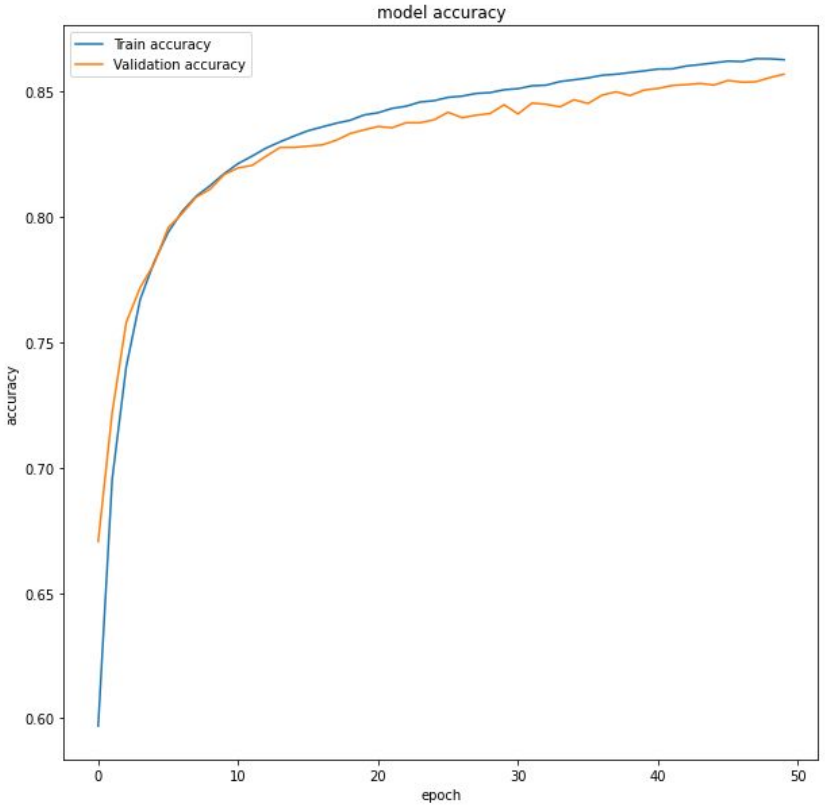


شکل ۲۷: ماتریس آشفتگی نرمال سازی نشده برای شبکه با batched_size=32

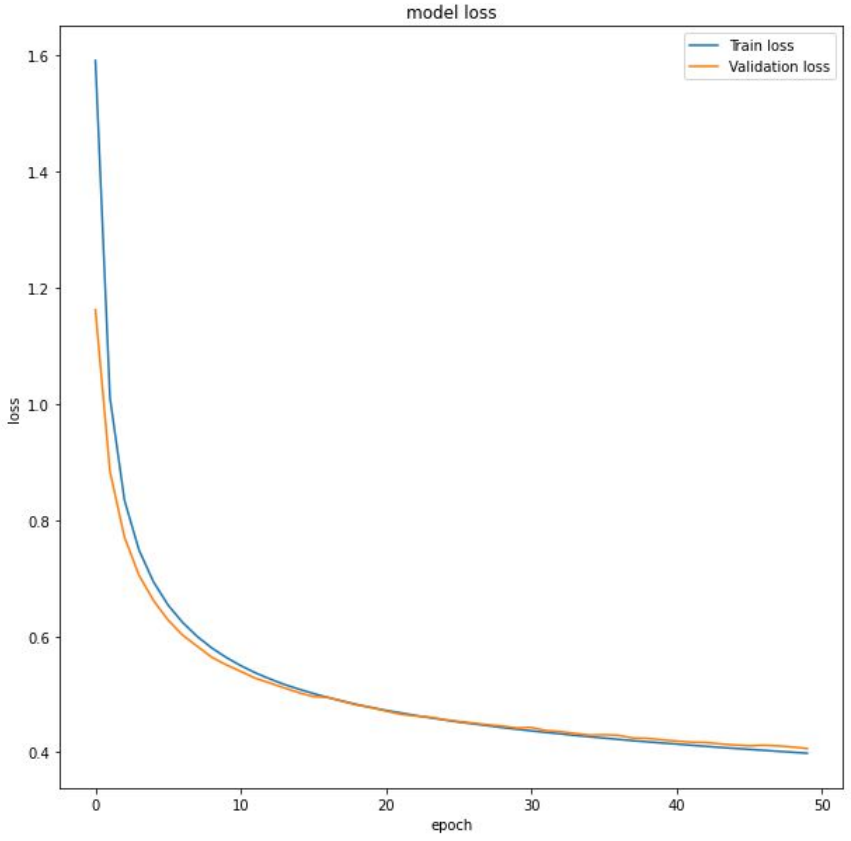


شکل ۲۸: ماتریس آشفتگی نرمال سازی شده برای شبکه با batched_size=32

در حالت batch size = 64 نمودارهای accuracy-epoch و loss-epochبه صورت زیر می باشد:



شکل ۲۹: نمودار accuracy-epoch برای شبکه با batched_size = 64

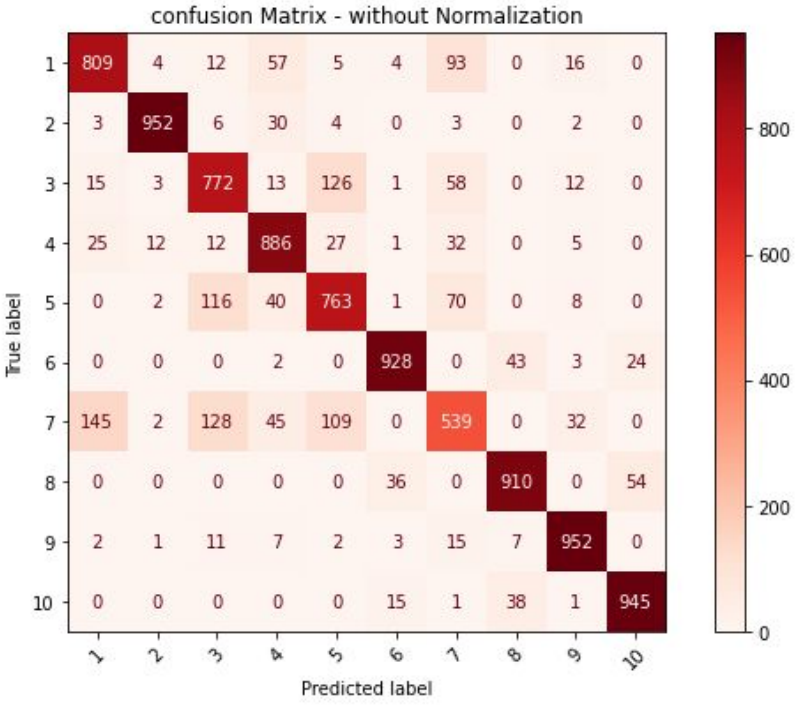


شکل ۳۰: نمودار loss-epoch برای شبکه با batched_size = 64

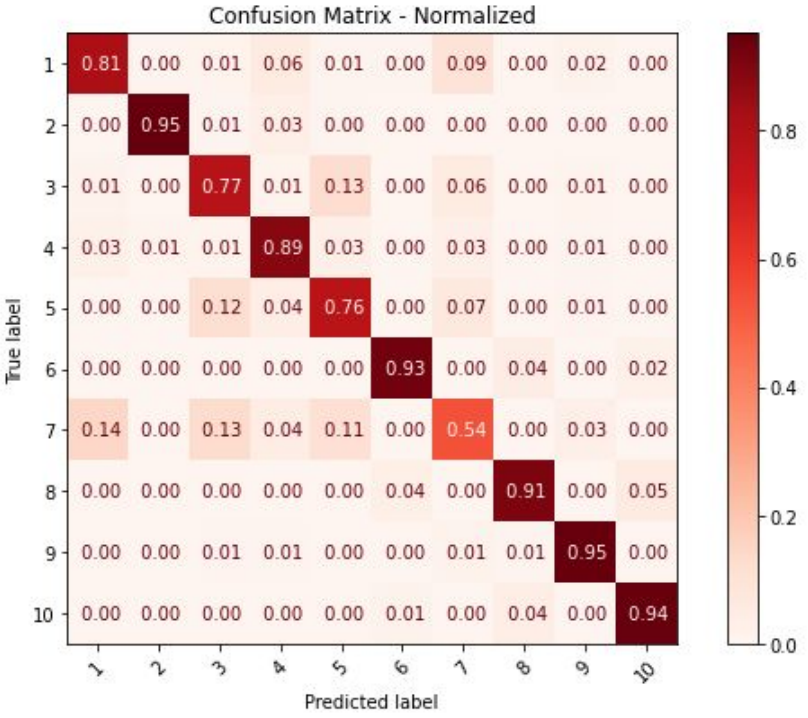
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.4350370466709137
accuracy in test data is :  0.8456000089645386
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:

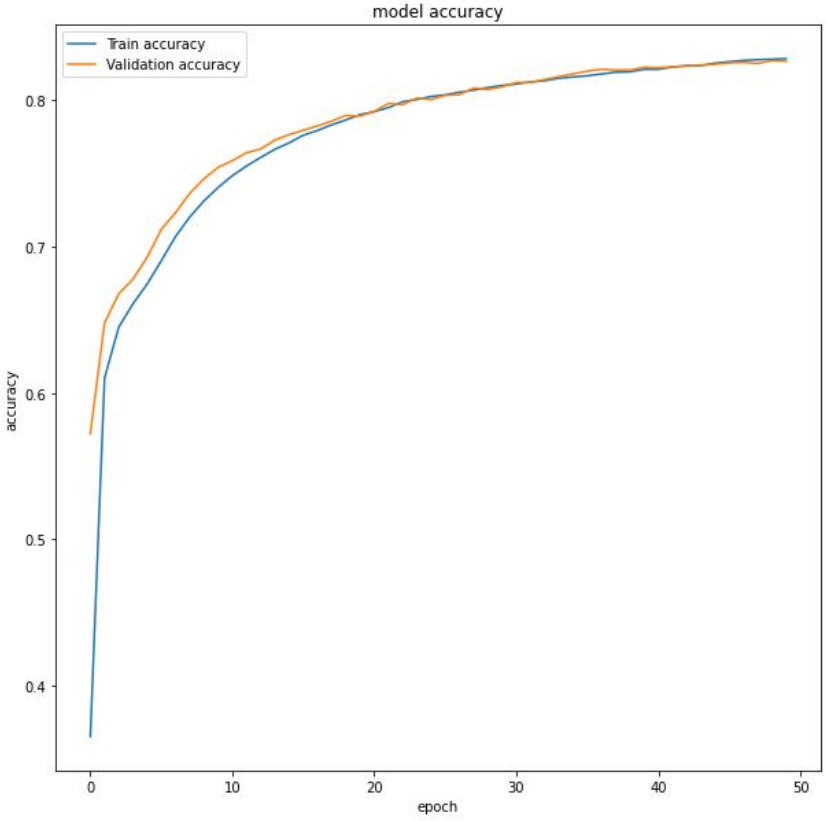


شکل ۳۱: ماتریس آشفتگی نرمال سازی نشده برای شبکه با batched_size=64

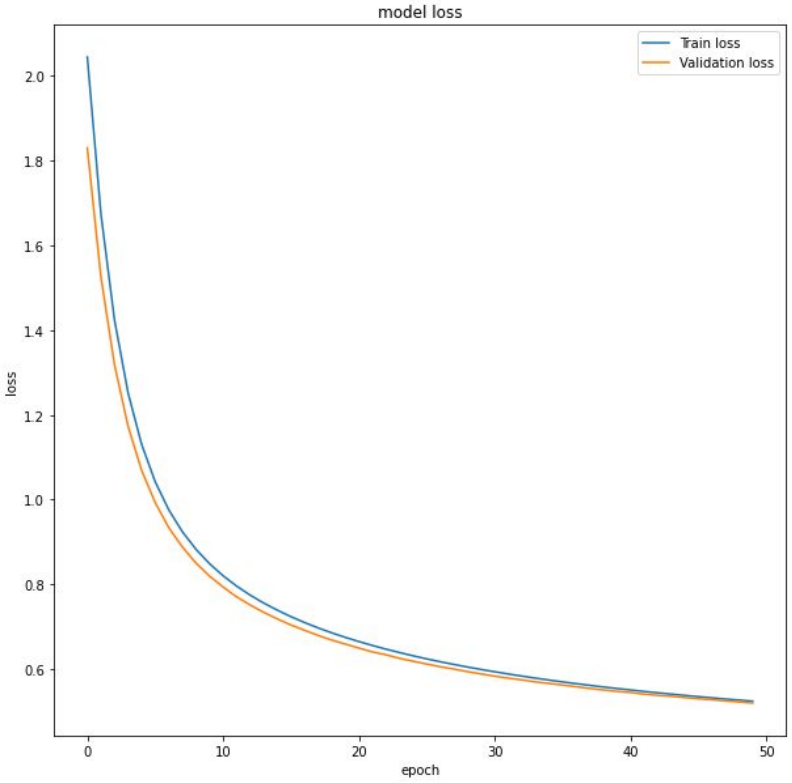


شکل ۳۲: ماتریس آشفتگی نرمال سازی شده برای شبکه با batched_size=64

در حالت batch size = 256 نمودارهای accuracy-epoch و loss-epochبه صورت زیر می باشد:



شکل ۳۳: نمودار accuracy-epoch برای شبکه با batched_size = 256

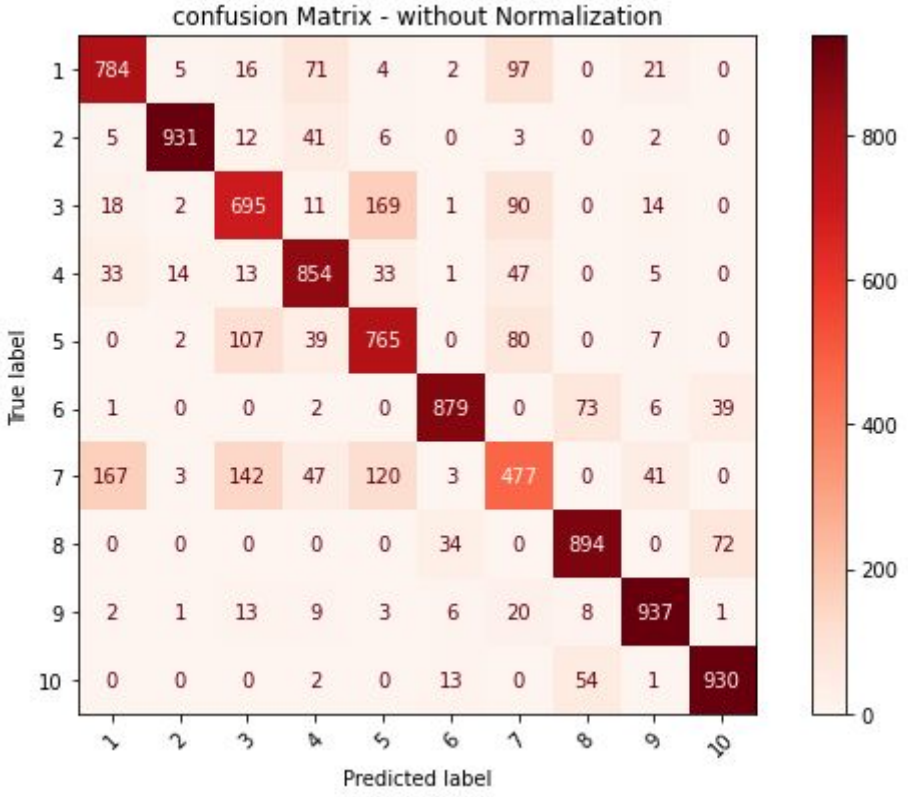


شکل ۳۴: نمودار loss-epoch برای شبکه با batched_size = 256

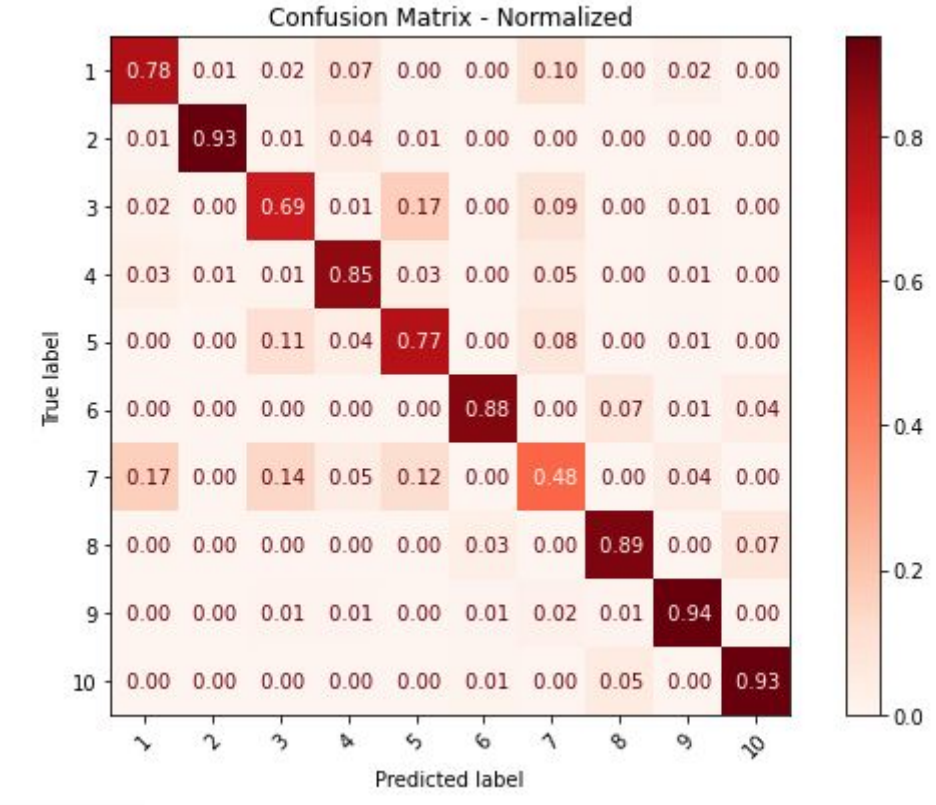
پس از اجرای داده های تست روی این شبکه ی آموزش دیده، میزان دقت و خطا برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.5449494123458862
accuracy in test data is :  0.817300021648407
```

همچنین ماتریس آشفتگی برای داده های تست به صورت زیر می باشد:



شکل ۳۵: ماتریس آشفتگی نرمال سازی نشده برای شبکه با batched_size=256



شکل ۳۶: ماتریس آشفتگی نرمال سازی شده برای شبکه با batched_size=256

از لحاظ دقت شبکه با batch_size = 32 دارای بیشترین دقت و شبکه با batch_size = 256 دارای کمترین دقت است و نیز شبکه با batch_size = 64 دقتی کمتر از شبکه با batch_size = 32 و دقتی بیشتر از شبکه با batch_size = 256 دارد.

از لحاظ مدت زمان آموزش شبکه، مدت زمان آموزش در شبکه با batch_size= 32 از زمان دو شبکه ی دیگر بیشتر و مدت زمان آموزش در شبکه با batch_size = 64 از مدت زمان آموزش در شبکه با batch_size = 256 بیشتر است.

پاسخ قسمت د) با انتخاب پارامتر ها به صورت زیر:

learning rate : 0.001

batch_size = 32

number of neurons in first layer : 1100

number of neuron in second layer : 900

epochs: 50

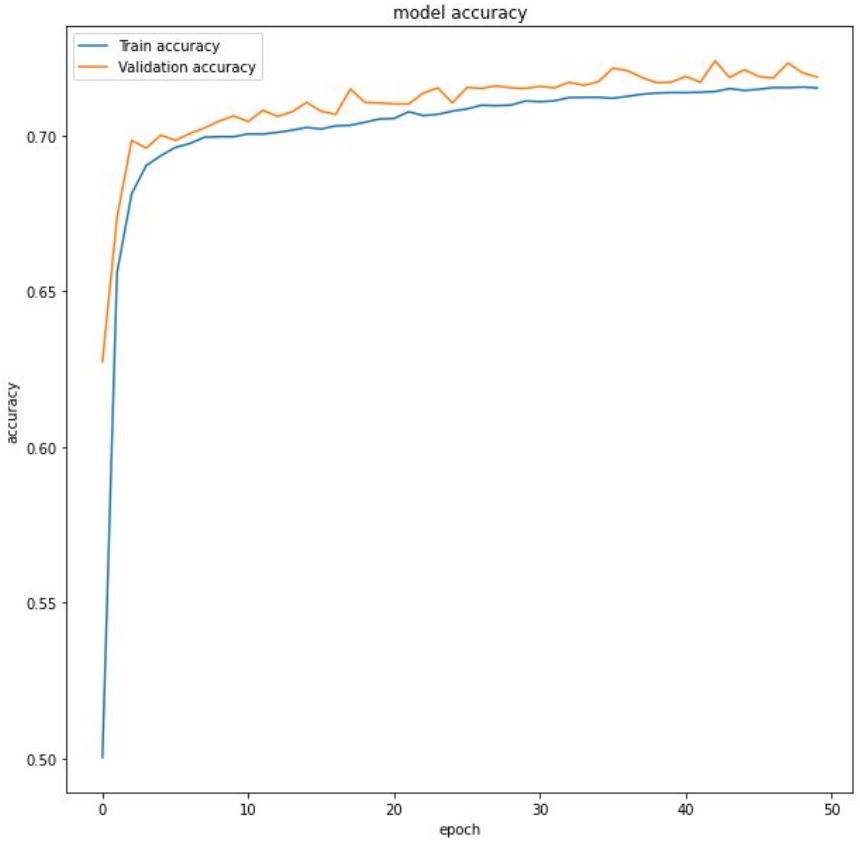
loss function : categorical_crossentropy

شبکه بهترین دقت نسبی معادل accuracy = 0.8618 را خواهد داشت.

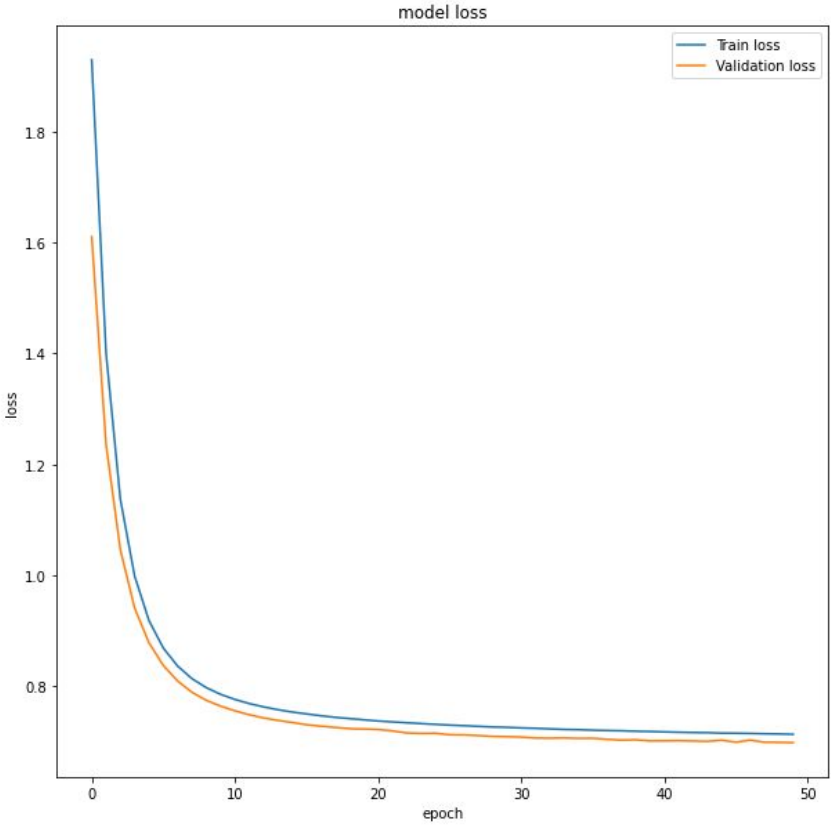
سوال 4 –Dimensionality Reduction

قسمت الف)

پس از کاهش بعد با این روش به ۵۰ ، نتایج به صورت زیر می باشد:



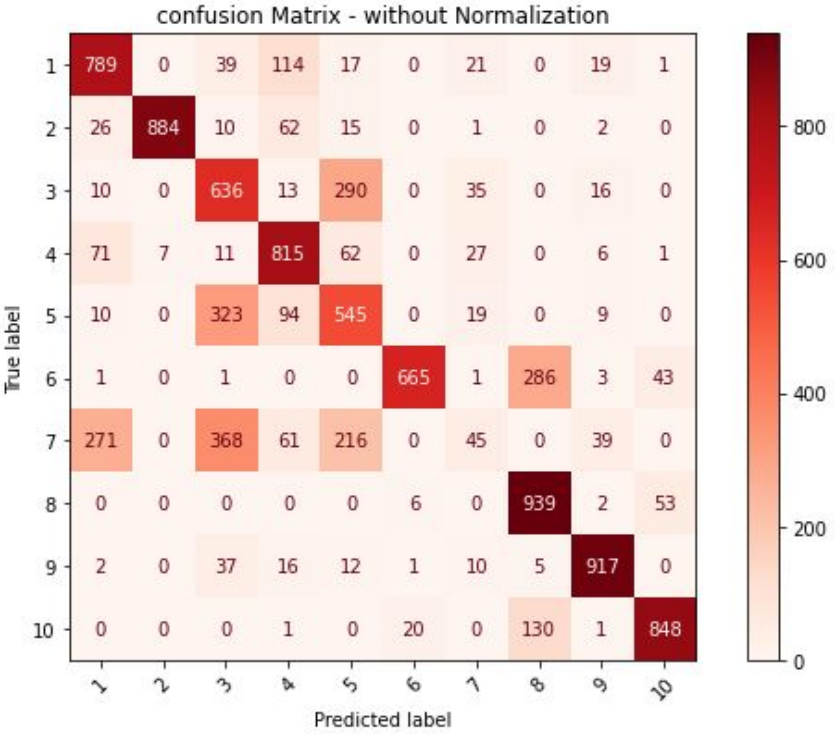
شکل ۳۷: نمودار accuracy-epoch



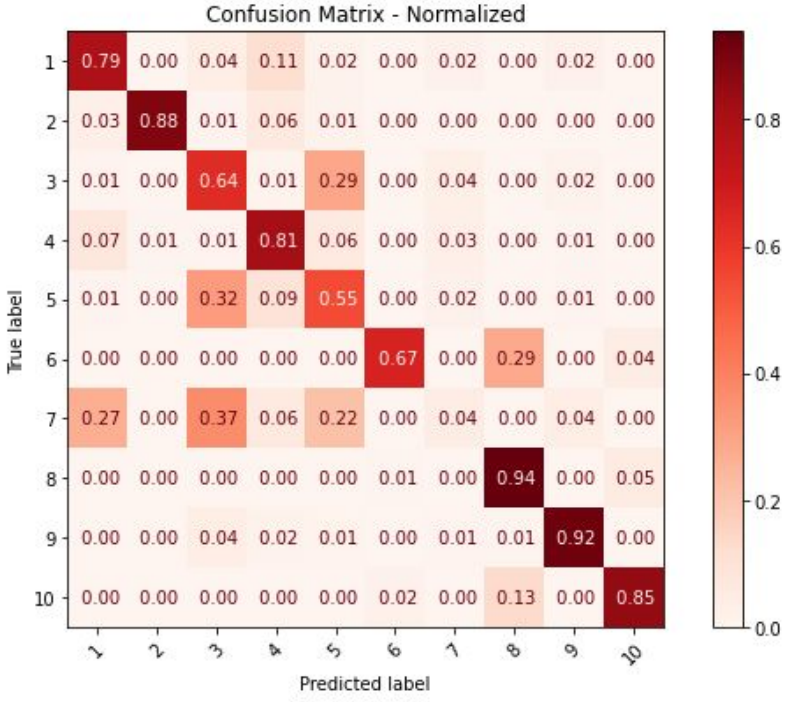
شکل ۳۸: نمودار loss-epoch

مقادیر دقت و خطا برای داده های تست در این شبکه به صورت زیر است:

```
loss in test data is:  0.731201708316803
accuracy in test data is :  0.708299994468689
```



شکل ۳۹: ماتریس آشفتگی نرمال سازی نشده

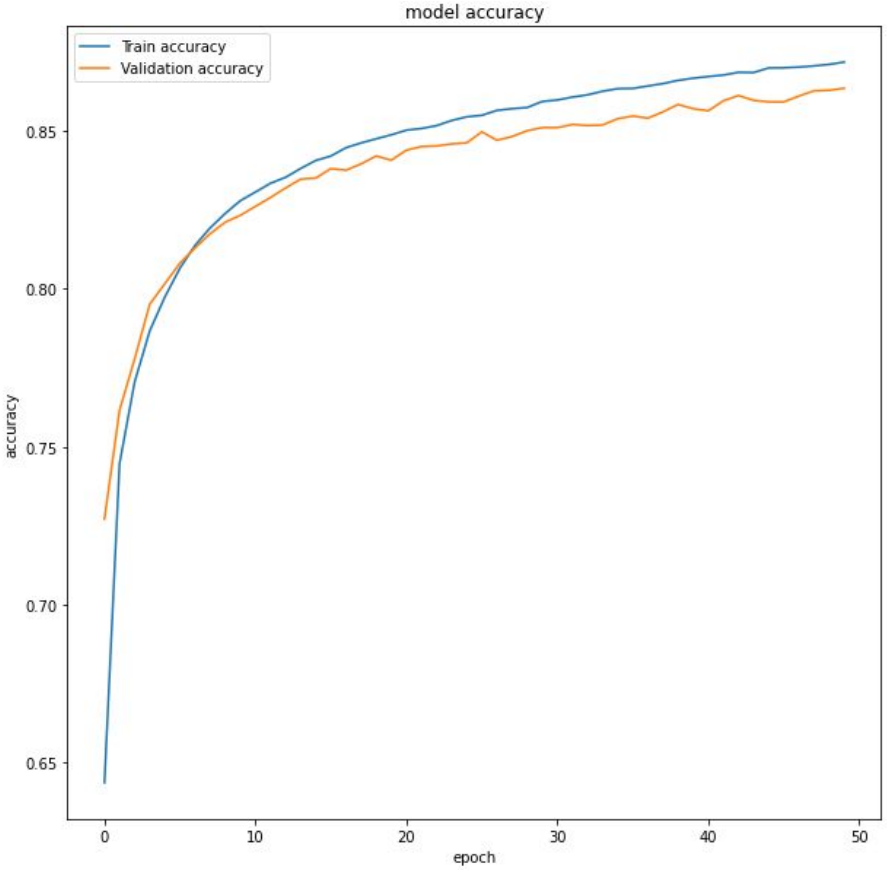


شکل ۴۰: ماتریس آشفتگی نرمال سازی شده

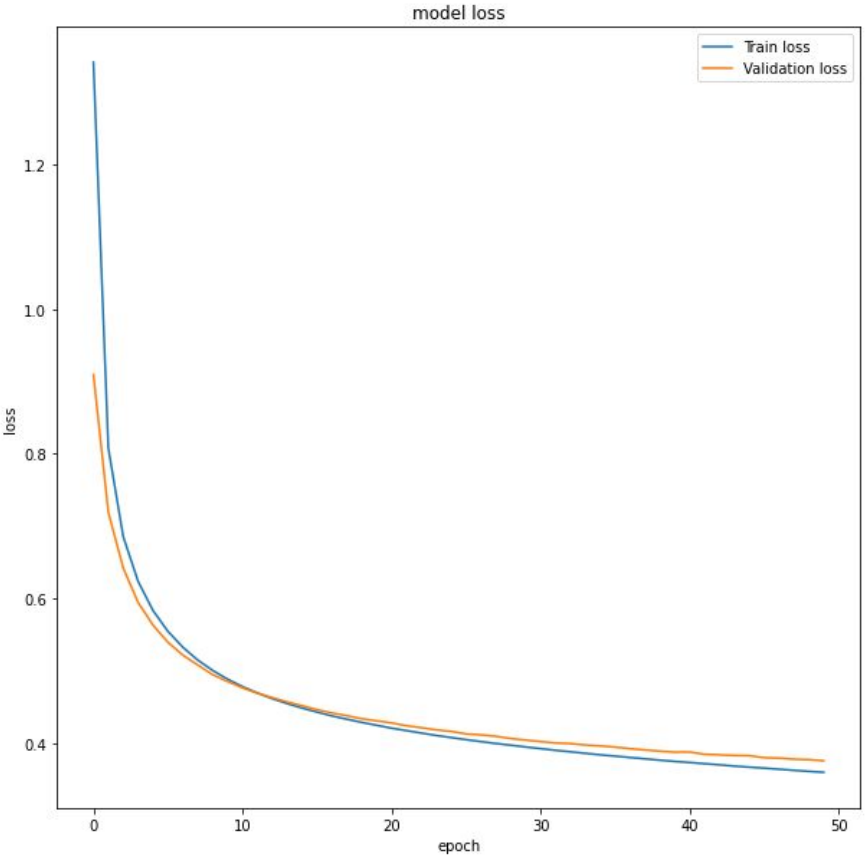
قسمت ب)

در روش کاهش بعد به کمک PCA، ابتدا الگوریتم جهت هایی که مجموعه داده پراکندگی بیشتری دارد را بدست می آورد سپس به وسیله ی این جهت ها بردارهای ویژه ماتریس را می سازد. پس از آن یا استفاده از یک تبدیل خطی پایه ای فضا را بر این بردارها منطبق می کند.

پس از کاهش بعد به این روش به ۵۰ نتایج به صورت زیر می باشد:



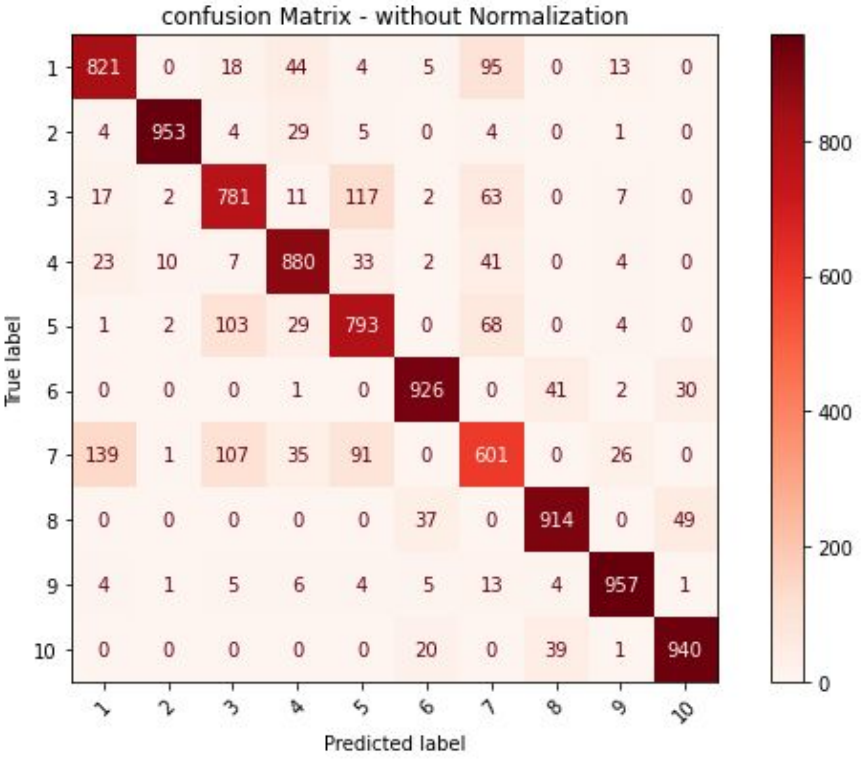
شکل ۴۱: نمودار accuracy-epoch



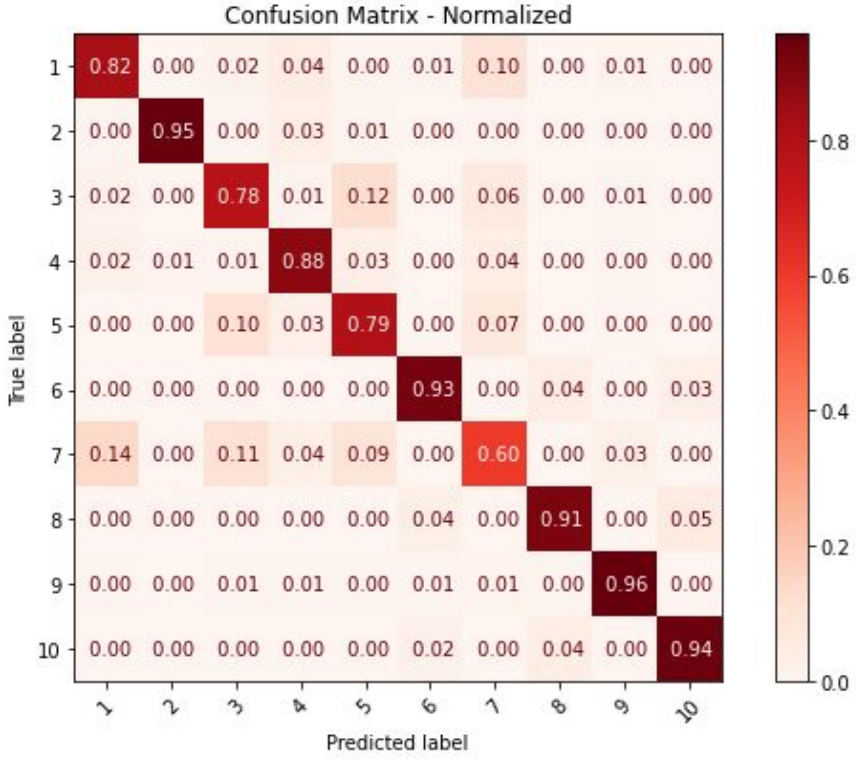
شکل ۴۲: نمودار loss-epoch

مقادیر دقت و خطا برای داده های تست در این شبکه به صورت زیر است:

```
loss in test data is:  0.40088769793510437
accuracy in test data is :  0.8565999865531921
```



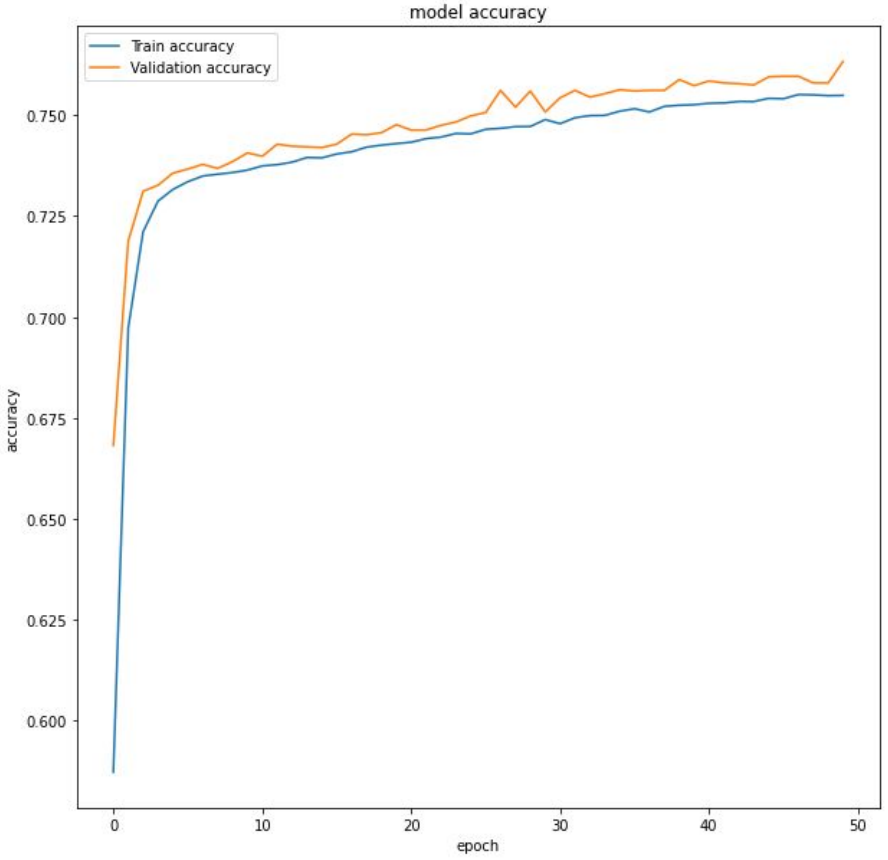
شکل ۴۳: ماتریس آشفتگی نرمال سازی نشده



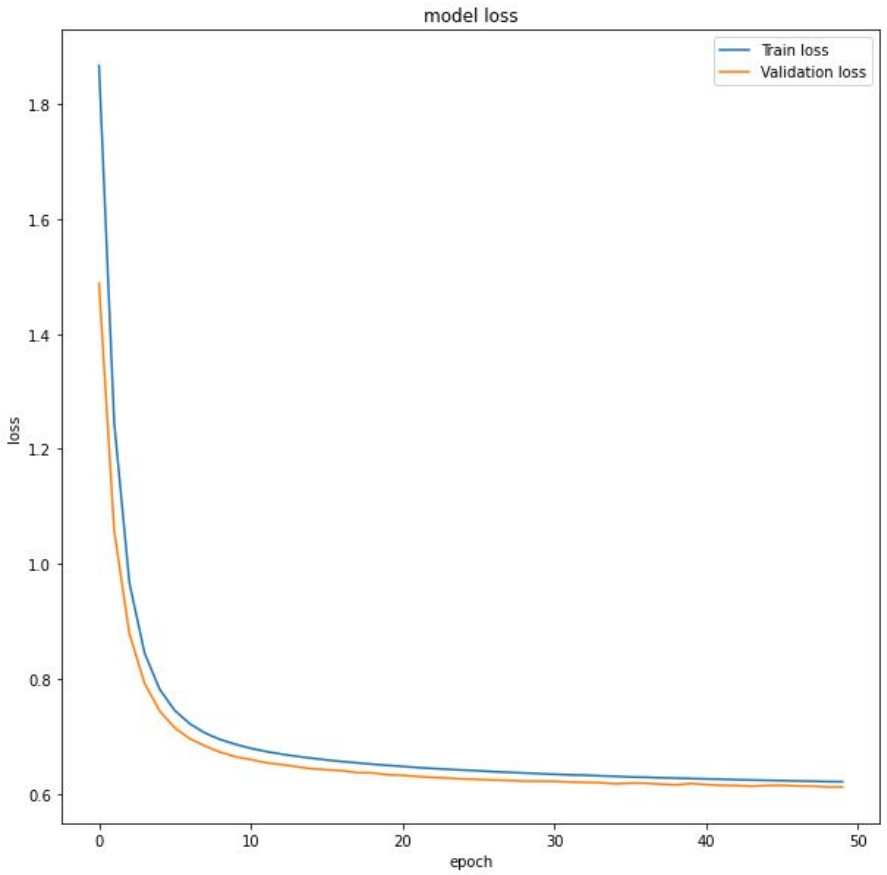
شکل ۴۴: ماتریس آشفتگی نرمال سازی شده

قسمت ج)

در این بخش نیز پس از کاهش بعد ابتدا به ۲۰۰ و سپس به ۵۰ نتایج به صورت زیر بدست آمد:



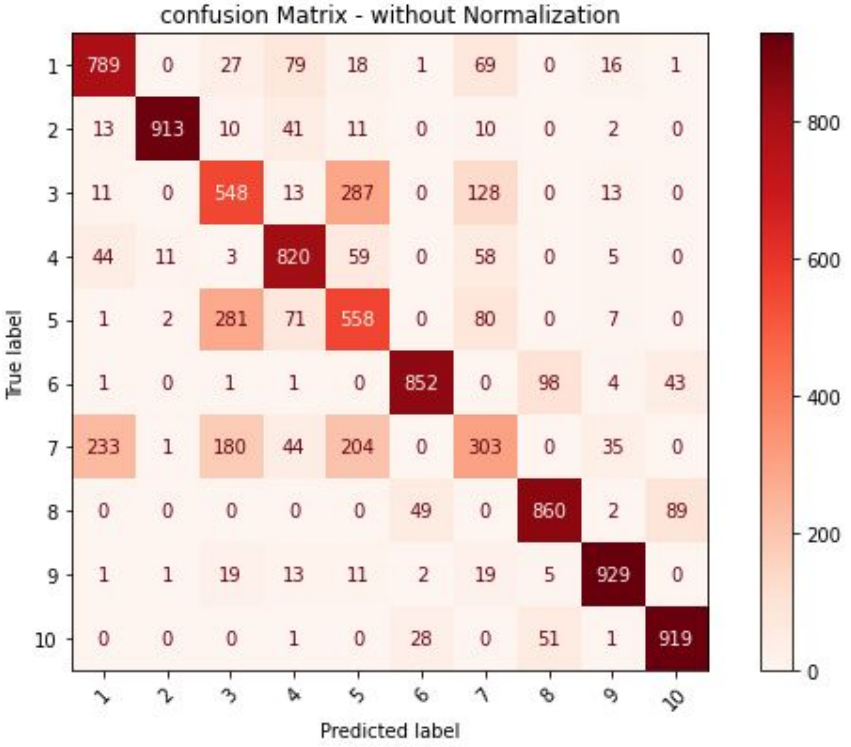
شکل ۴۵: نمودار accuracy-epoch



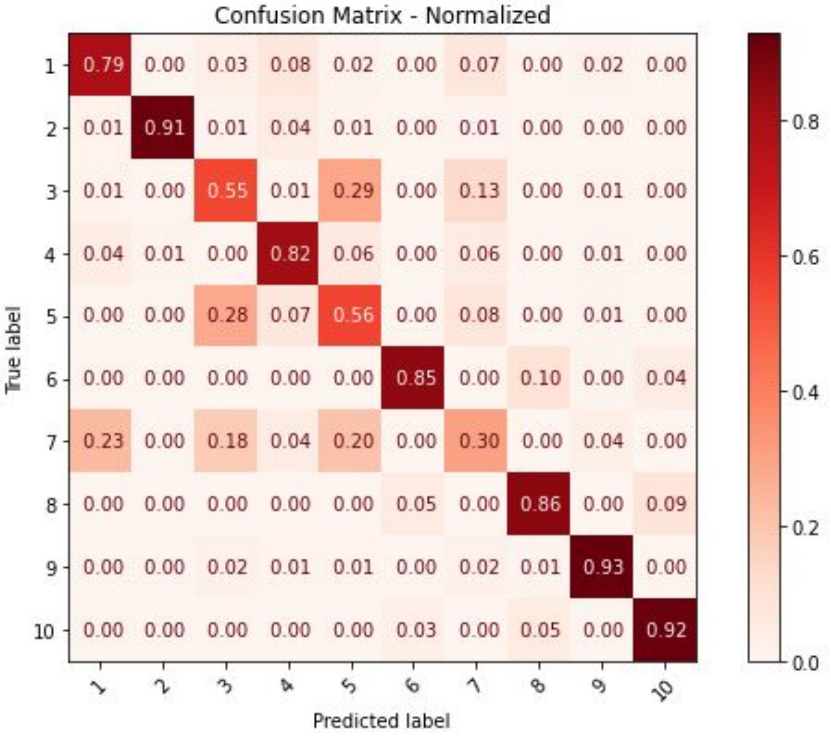
شکل ۴۶: نمودار loss-epoch

همچنین مقدار خطا و دقت شبکه برای داده های تست به صورت زیر می باشد:

```
loss in test data is:  0.6367365121841431
accuracy in test data is :  0.7491000294685364
```



شکل ۴۷: ماتریس آشفتگی نرمال سازی نشده



شکل ۴۸: ماتریس آشفتگی نرمال سازی شده

قسمت د)

جدول(۱)- مقایسه ی دقت شبکه های مختلف

مورد	دقت داده های تست	خطا ی داده های تست
بهترین شبکه ی بدست آمده در سوال ۳	0.8618	0.3940
AutoEncode	0.7082	0.7312
PCA	0.8586	0.0400
Cascade RBM	0.7491	0.6367

به طور کلی از کاهش بعد برای از بین بردن فضای پوچی استفاده می کنیم بدون آنکه اطلاعات مفید و مورد نیاز ما از بین برود. به طوری با عکس این عمل بتوانیم به بعد اولیه برسیم.

به طور کلی در روش کاهش بعد به وسیله ی autoEncoder کاهش بعد و از بین بردن فضای پوچی نسبت به روش cascaded RBM بهتر صورت می گیرد و autoEncoder قابلیت کاهش بعدبرای نگاشت های پیچیده تر نسبت به cascaded RBMرا داراست. کاهش بعد به وسیله ی روش PCA، روش مناسبی می باشد اما این مسئله را باید رد نظر بگیریم که از PCA تنها می توان برای کاهش بعد خطی استفاده کرد.

از مقایسه ی نتایج بدست آمده طبق جدول بالا می توان گفت، نتیجه ی بدست آمده در سوال سوم(بدون استفاده از ماهش بعد بر روی داده ها) بهتر است. علت حاصل شدن این نتیجه آن است که در ساختار سوال ۳ هر لایه بدون وجود محدودیت بر روی داده ها آموزش داده شدند و علاوه بر کاهش بعد داده، لایه ها classification نیز انجام می دهند و بطور مقایسه ای این شبکه نتیجه ی بهتری دارد.

سوال 5 –مفاهیم

قسمت الف)

در صورتی که تعداد داده های موجود در هر دسته برابر نباشد، شبکه ما بر اساس کلاسی که داده های آن بیشتر است آموزش می بیند به عبارت دیگر چون داده های یک دسته بیشتر است، و شبکه آن ها را بیشتر هنگام آموزش می بیند به روز رسانی های وزن ها و مقادیر بایاس را بیشتر بر اساس آن داده ها انجام می دهد و هنگام اجرای شبکه بر روی داده های تست، بیشتر داده ها را از نوع کلاس با تعداد داده ی بیشتر تشخیص می دهد. بنابراین طبقه بندی به کمک این شبکه به درستی صورت نمی گیرد.

راه حل این مشکل آن است که از loss function هایی استفاده کنیم که تعداد داده های هر کلاس را نیز در نظر می گیرند.

قسمت ب)

خیر- در صورتی که دقت یک شبکه بر روی داده های تست از شبکه های دیگر بیشتر باشد، لزوما نمی توان گفت که شبکه مناسب تر می باشد. زیرا شبکه ها را نسبت به یک مجموعه بررسی می کنیم و ممکن است به علت overfitting نتیجه ی دقت یک شبکه بر روی آن مجموعه داده نسبت به بقیه ی شبکه ها بهتر باشد اما لزوما نمی توان گفت که آن شبکه نسبت به بقیه ی شبکه ها تعمیم پذیری بهتری دارد. شبکه ای بهتر است که همواره قدرت تعمیم دهی آن از بقیه ی شبکه ها بیشتر باشد نه اینکه همواره دقت آن بر روی یک مجموعه داده بیشتر باشد.

قسمت ج)

به طور کلی برای آموزش یک شبکه عصبی می توان ویژگی هایی را انتخاب کرد که برای حل مشکل ما مفید تر و تاثیرگذار تر هستند. به این فرآیند انتخاب ویژگی یا feature selection می گویم.

انتخاب ویژگی همچنین انتخاب متغیر یا انتخاب ویژگی ها نامیده می شود.

سه الگوریتم انتخاب ویژگی وجود دارد: روش های فیلتر (**Filter Methods**) ، روش های بسته بندی (**Wrapper Methods**) و روش های تعبیه شده(**Embedded Methods**).

روش های انتخاب ویژگی فیلتر از یک معیار آماری برای تعیین امتیاز به هر ویژگی استفاده می کنند. این ویژگی ها با نمره رتبه بندی می شوند و برای نگهداری یا حذف از مجموعه داده انتخاب می شوند. روشها غالباً یک متغیر هستند و ویژگی را مستقل یا با توجه به متغیر وابسته در نظر می گیرند.

روش های بسته بندی هنگامی که ترکیب های مختلفی ایجاد ، ارزیابی و با یکدیگر مقایسه می شوند مجموعه ای از ویژگی ها را به عنوان یک search problem در نظر می گیرد .از مدل پیش بینی شده ما برای ارزیابی ترکیبی از ویژگی ها و تعیین نمره بر اساس دقت مدل استفاده می شود. فرآیند جستجو ممکن است methodical باشد مانند بهترین جستجو ، ممکن است تصادفی باشد مانند الگوریتم تصادفی hill-climbing ، یا ممکن است از اکتشافی مانند گذرهای رو به جلو و عقب برای اضافه کردن و حذف ویژگیها استفاده کند.

روش های تعبیه شده یاد می گیرند که هنگام ایجاد مدل، کدام ویژگی ها بهترین دقت را وجود می آورند. متداول ترین نوع روشهای انتخاب ویژگی embedded ، روشهای regularization می باشد.

قسمت د)

ماتریس آشفتگی روشی برای خلاصه کردن عملکرد یک الگوریتم طبقه بندی است. اگر تعداد مشاهده های نابرابر را در هر کلاس داشته باشیم یا اگر بیش از دو کلاس در مجموعه داده های خود داریم دقت طبقه بندی به تنهایی می تواند

گمراه کننده باشد. محاسبه یک ماتریس آشفتگی می تواند به ما ایده بهتر دهد که مدل طبقه بندی مت به درستی انجام می شود و چه نوع خطایی را ایجاد می کند. بنابراین به کمک ماتریس آشفتگی می توان انواع مختلفی از خطا ها را بدست آورد. همچنین به وسیله ی normalization های متفاوت توسط این ماتریس می توان توصیفات مختلفی از نتیجه ی آن داشت.

همچنین می توانیم از این ماتریس برای ایجاد confidence matrix استفاده کرد

قسمت ه)

نرمال سازی و استاندارد سازی، دو روش مقیاس پذیر مورد بحث می باشد. عادی سازی یا نرمال سازی به طور معمول به معنای ذخیره مقادیرداده در محدوده [0،1] است. این تکنیک ممکن است در بعضی موارد مفید باشد که همه پارامترها دارای مقیاس مثبت یکسانی باشند. اما استانداردسازی به طور معمول به معنای ذخیره داده ها با میانگین 0 و انحراف استاندارد 1 (واریانس واحد) است.