به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

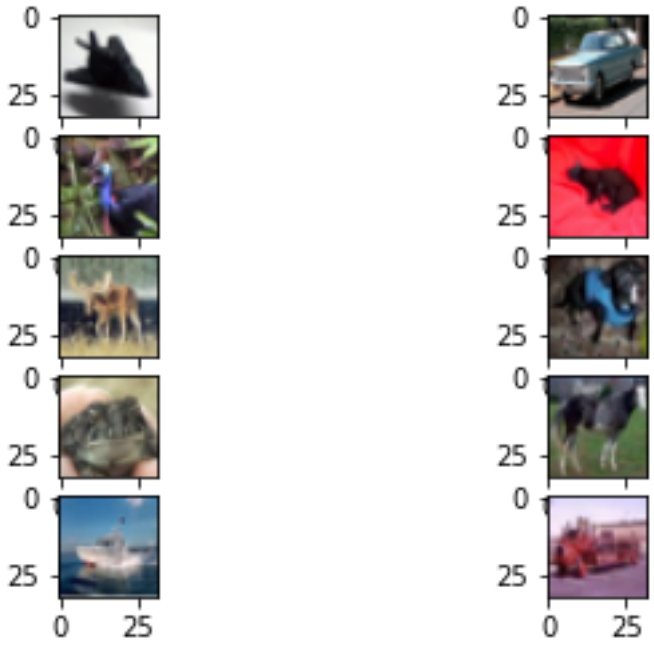
درس شبکه‌های عصبی و یادگیری عمیق

تمرین سری ۲

نام و نام خانوادگی	سیاوش شمس
شماره دانشجویی	810197644
تاریخ ارسال گزارش	۰۱/۰۱/۱۹

فهرست گزارش سوالات

سوال ۱ – MLP(Classification)



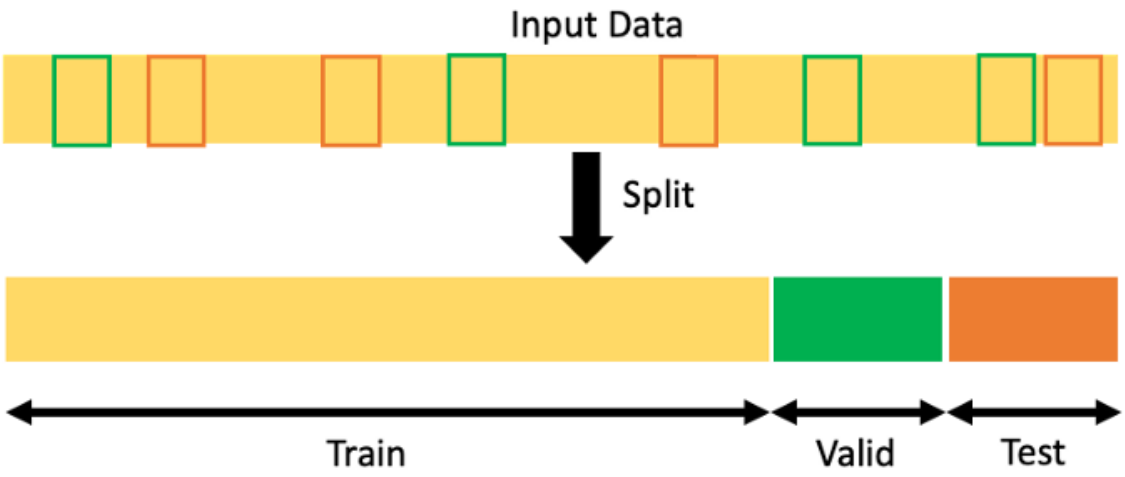
شکل ۱-۱ ۱۰ تصویر مختلف از دادگان

الف)

برای جداسازی داده های آموزش، ارزیابی و تست بسته به نوع مسئله روش های مختلفی داریم:

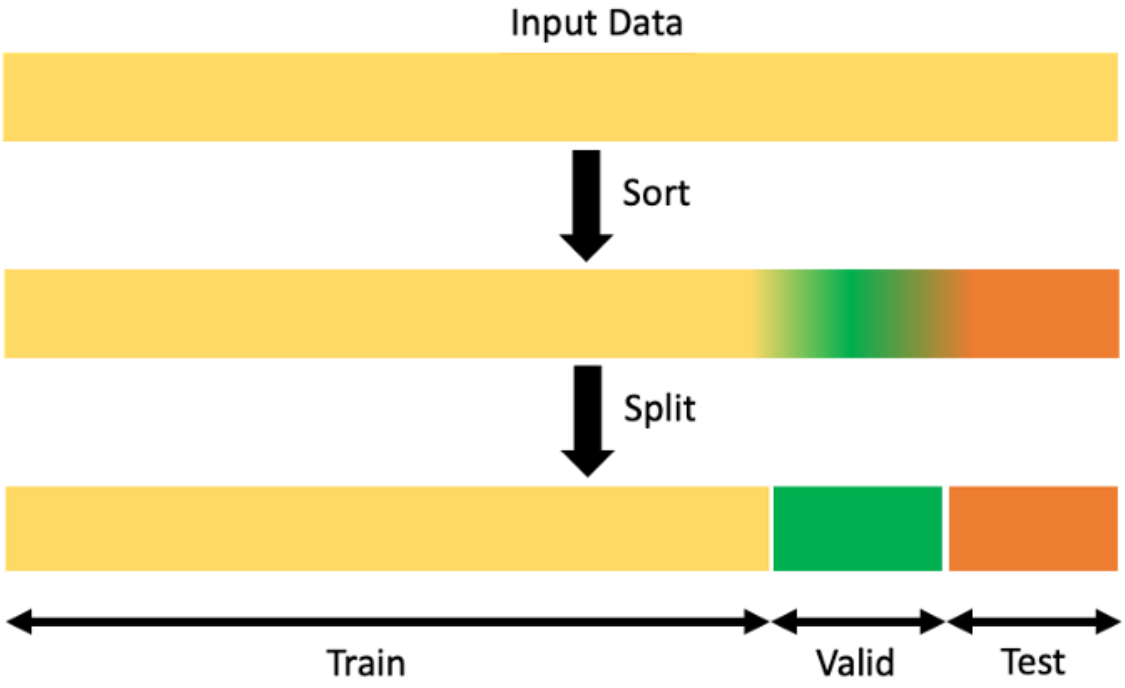
۱. جداسازی به صورت تصادفی
۲. جداسازی به صورت جزء زمانی

در روش اول که متداول تر است داده های مخصوص آموزش، ارزیابی و تست به صورت تصادفی با نسبت های مربوطه تقسیم می شوند، شکل ۱-۲ درک شهودی از این روش تقسیم را به ما می دهد.



شکل ۱-۲ جداسازی به صورت تصادفی

در روش دوم که بیشتر برای پیشبینی وضعیت آینده کاربرد دارد، دادها با توجه به سری زمانی موجود تقسیم بندی می شوند، به عبارت دیگر داده ها ابتدا بر اساس پارامتر مورد نظر مرتب می شوند و بعد داده ها تا زمان خاصی برای آموزش مورد استفاده قرار می گیرد و داده های پس از آن زمان برای ارزیابی و تست مورد استفاده قرار می گیرند. شکل ۱-۳ درک شهودی از این روش تقسیم به ما می دهد



شکل ۱-۳ جداسازی به صورت جزء زمانی

برای مشخص کردن نسبت تقسیم داده ها باید به تعداد کل داده ها و مدلی که آموزش می دهیم توجه شود، برای مثال در بعضی از موارد نیاز به داده های بیشتری برای آموزش نیاز داریم، یا در بعضی

موارد فراپارامتر[1] های کمی برای آموزش نیاز است در این موارد می توان بخش کمتری را به داده های مربوط به ارزیابی اختصاص داد.

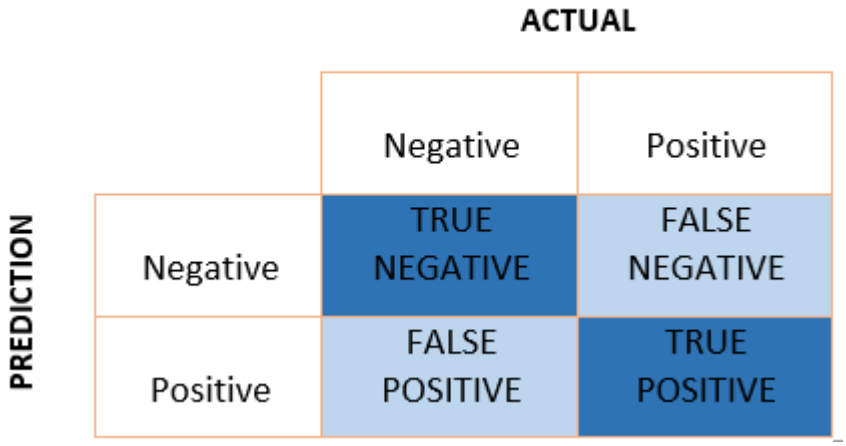
ولی به طور معمول از نسبت 10-10-80 برای تقسیم داده های آموزش، ارزیابی و تست استفاده می شود.

منبع:

https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c

ب)

ماتریس آشفتگی[2] به ماتریسی گفته می‌شود که در آن عملکرد الگوریتم‌های مربوطه را نشان می‌دهند. روی یک محور تعداد داده های هر دسته با برچسب واقعی آنها قرار دارند و روی محور دیگر تعداد داده ها با برچسب پیشبینی شده قرار دارند. در شکل ۱-۴ نمونه یک ماتریس آشفتگی 2 × 2 را می بینیم.



شکل ۱-۴ ماتریس آشفتگی 2 × 2

TP بیانگر تعداد داده هایی است که برچسب مثبت آن ها به درستی پیش بینی شده است

FN بیانگر تعداد داده هایی است که برچسب منفی آن ها به غلط پیش بینی شده است

FP بیانگر تعداد داده هایی است که برچسب مثبت آن ها به غلط پیش بینی شده است

TN بیانگر تعداد داده هایی است که برچسب منفی آن ها به درستی پیش بینی شده است

[1] Hyperparameter
[2] Confusion Matrix

دقت (Precision) بیانگر نسبت تعداد برچسب های مثبت که به درستی پیش بینی شده اند به کل داده هایی که مثبت پیش بینی شده اند می باشد.

$$\text{Precision} = \frac{TP}{TP + FP}$$

فراخوانی (Recall) بیانگر نسبت تعداد برچسب های مثبت که به درستی پیش بینی شده اند به کل تعداد داده ها با برچسب مثبت می باشد.

$$\text{Recall} = \frac{TP}{TP + FN}$$

معیار F1 (F1 score) در واقع ترکیب دقت و فراخوانی می باشد، این معیار برای دادگان های غیر متعادل معیار مناسبی است که به صورت زیر به دست می آید

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

منبع:

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

ت)

به عنوان پیش پردازش:

- داده ها را نرمالایز کرده
- برچسب آن ها را به صورت one-hot کد می کنیم.
- تصاویر را از RGB به Grayscale تبدیل کردیم
- تغییر ابعاد تصاویر از 32 × 32 به 1024 تا قابلیت فید شدن به شبکه ANN را داشته باشد

از الگوریتم SGD با mini batch برای به دست آوردن پارامتر ها استفاده کردیم. از تابع فعالساز ReLU به عنوان فعالساز برای هر لایه مخفی و از تابع فعالساز Softmax برای لایه خروجی استفاده کردیم، سایز هر بسته را ۶۴ در نظر گرفتیم. مدل را در ۲۰ تکرار آموزش می دهیم

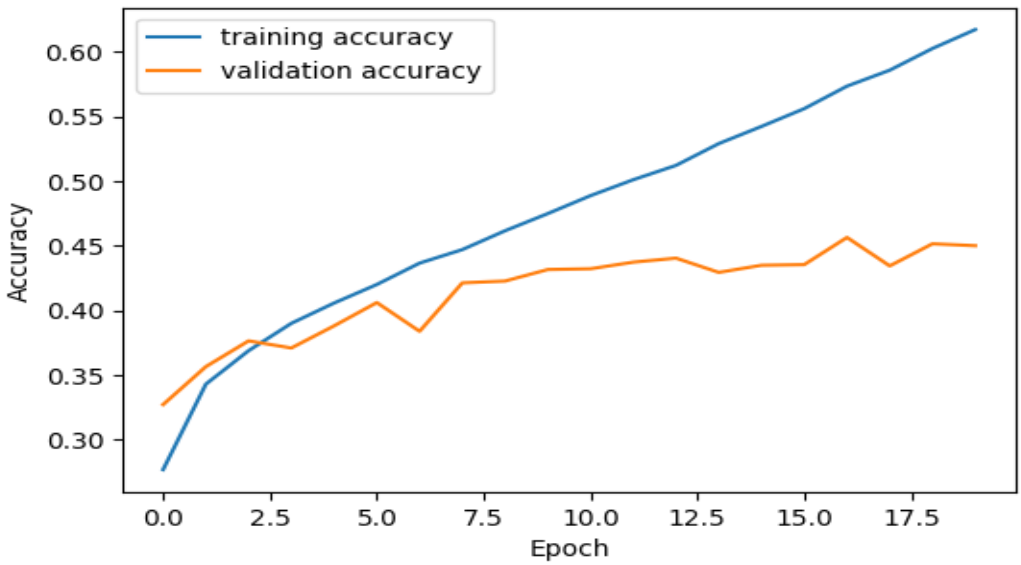
جدول ۱-۱ معماری شبکه برای قسمت ت

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_37 (Dense)            (None, 1024)              1049600

 dense_38 (Dense)            (None, 1024)              1049600

 dense_39 (Dense)            (None, 512)               524800

 dense_40 (Dense)            (None, 10)                5130

=================================================================
Total params: 2,629,130
Trainable params: 2,629,130
Non-trainable params: 0
_________________________________________________________________
```

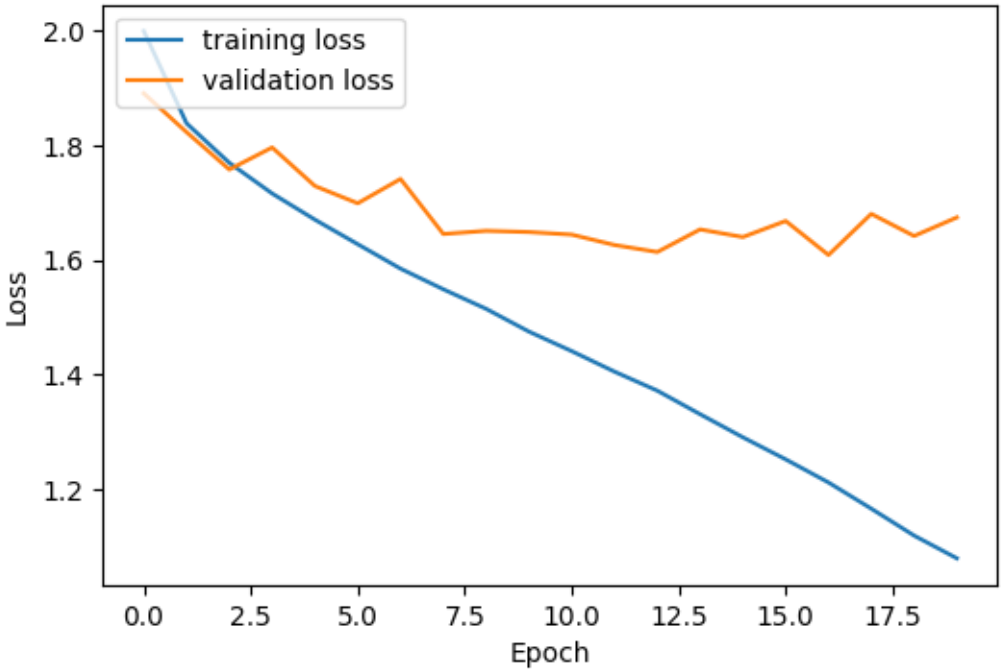
با توجه به معماری بالا که دارای دو لایه تماما متصل با ۱۰۲۴ نورون و سپس یک لایه تماما متصل با ۵۱۲ نورون به عنوان لایه های مخفی می باشد نتایج زیر را به دست می آوریم.

```
training time: 86.24829077720642
```

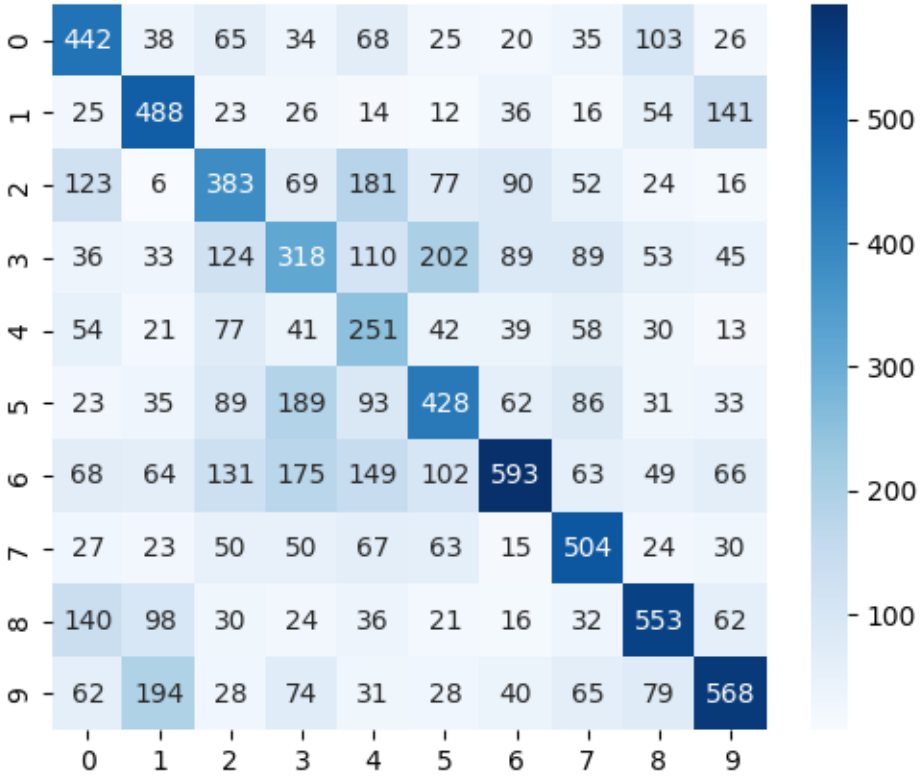
شکل ۱-۵ زمان آموزش مدل



شکل ۱-۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۸ ماتریس آشفتگی

```
accuracy on test data: 0.4528000056743622
loss on test data: 1.6581695079803467
f1_score on test data: 0.4512196455685049
recall on test data: 0.4528
presicion on test data: 0.45959435594477716
```

شکل ۱-۹ معیار های مختلف روی داده های تست

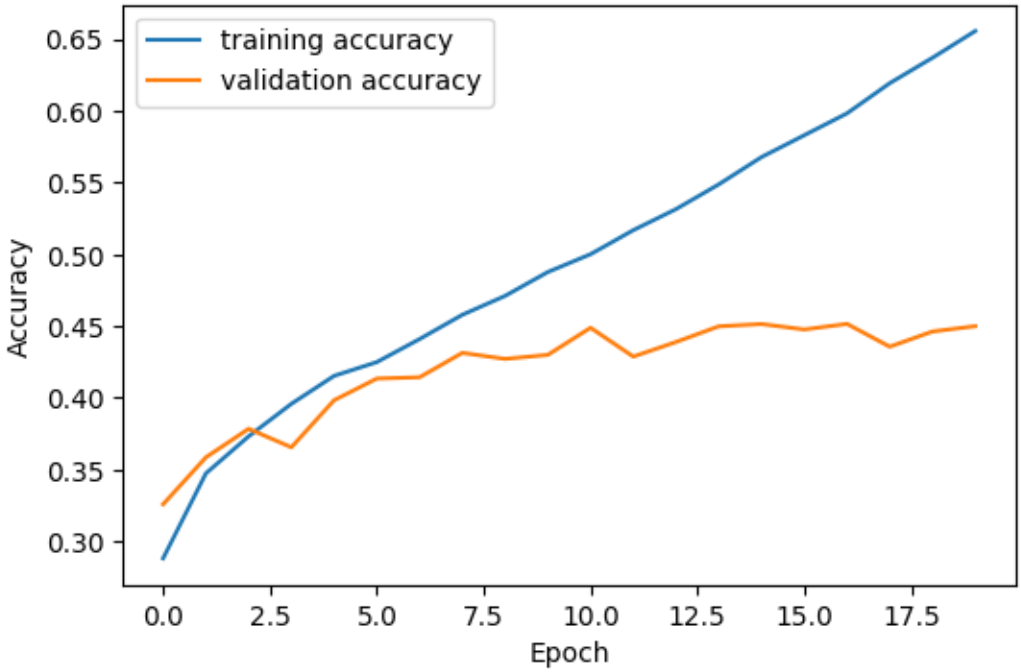
حال تعداد نورون های لایه های تماما متصل را دو برابر می کنیم.

جدول ۱-۲ معماری شبکه

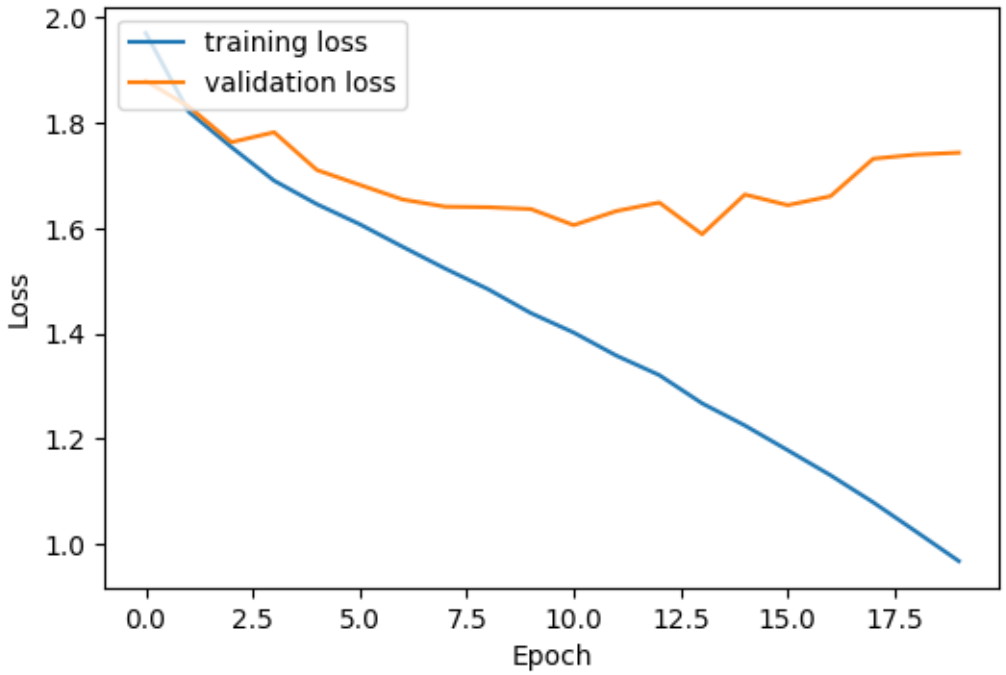
```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_45 (Dense)            (None, 2048)              2099200

 dense_46 (Dense)            (None, 2048)              4196352

 dense_47 (Dense)            (None, 1024)              2098176

 dense_48 (Dense)            (None, 10)                10250

=================================================================
Total params: 8,403,978
Trainable params: 8,403,978
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 143.3704309463501
```

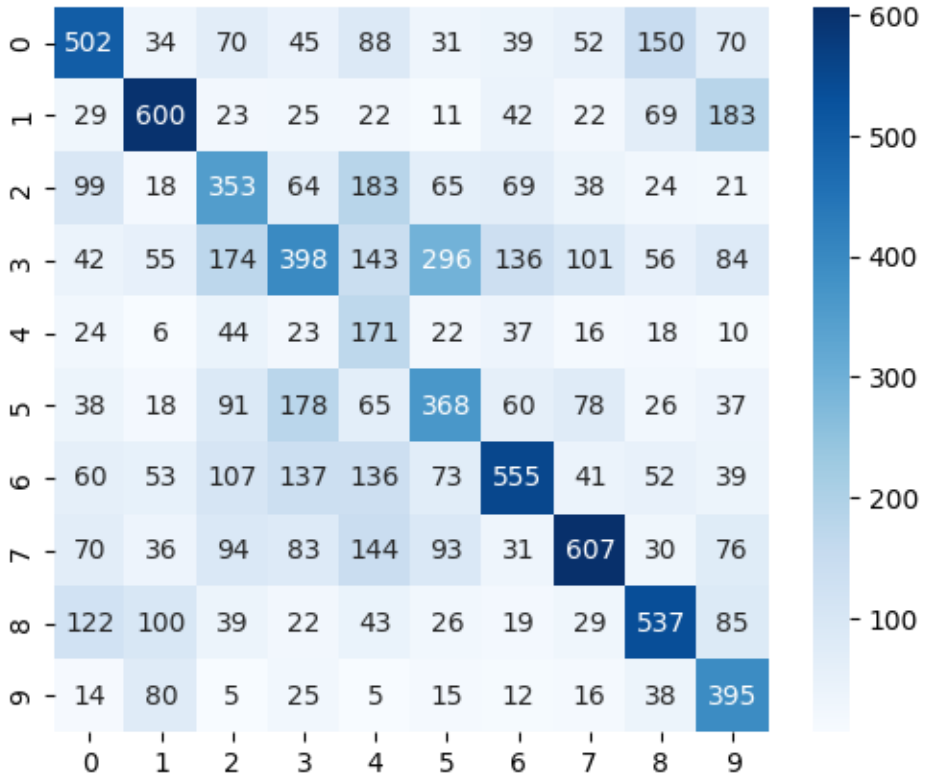
شکل ۱-۱۰ زمان آموزش مدل



شکل ۱-۱۱ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۲ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۳ماتریس آشفتگی

```
accuracy on test data: 0.44859999418258667
loss on test data: 1.7459070682525635
f1_score on test data: 0.44375483850814723
recall on test data: 0.4486
presicion on test data: 0.46412787867084765
```

شکل ۱-۱۴ معیار های مختلف برای داده تست

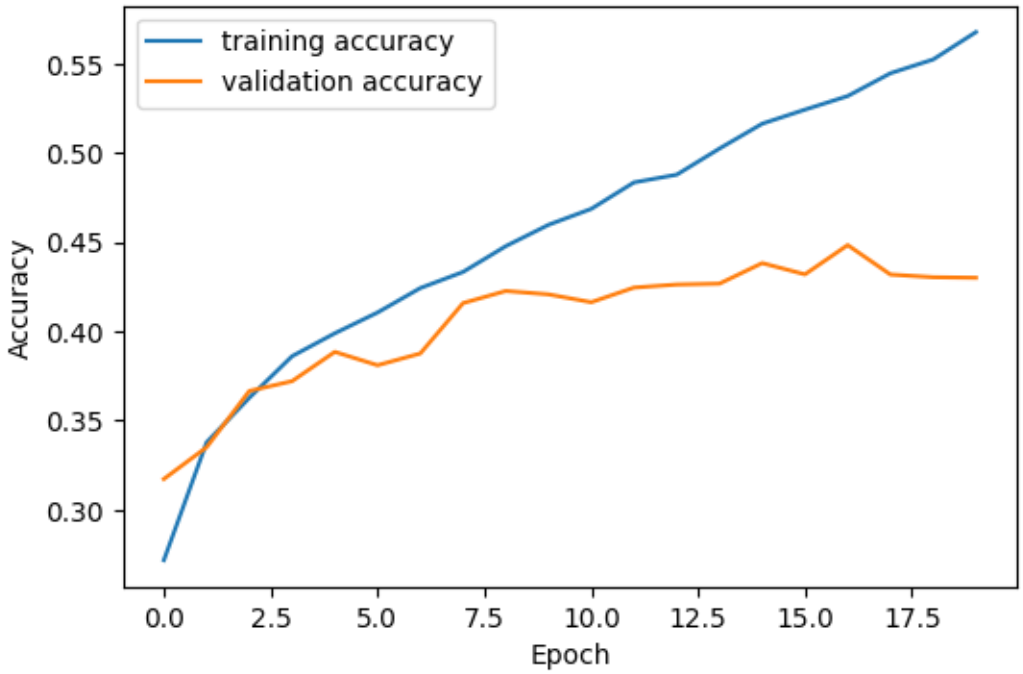
حال تعداد نورون های لایه های تماما متصل را نصف می کنیم.

جدول ۱-۳ معماری مدل

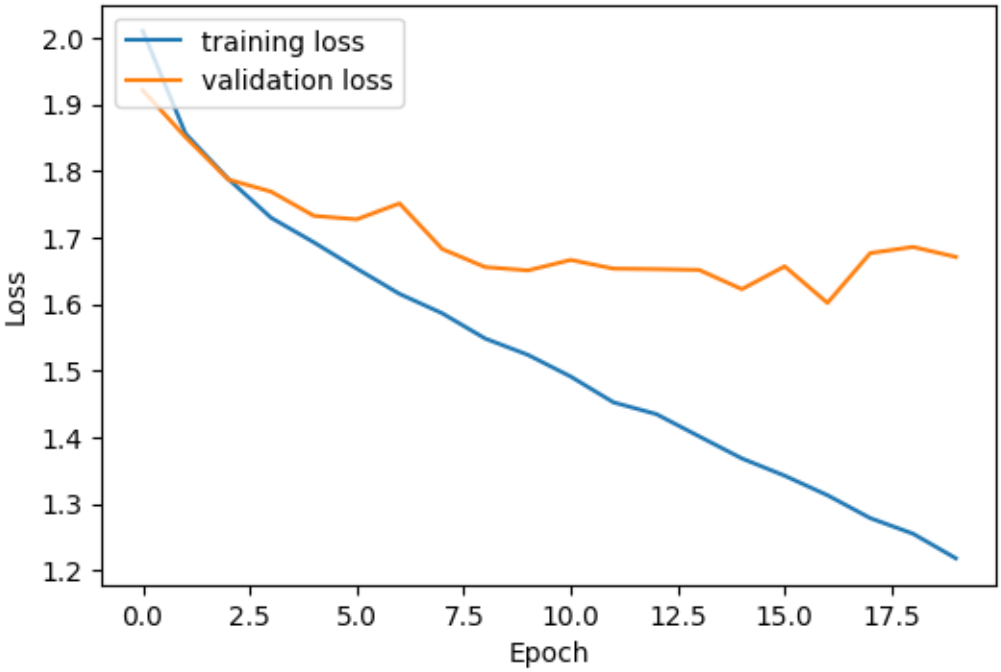
```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_41 (Dense)            (None, 512)               524800

 dense_42 (Dense)            (None, 512)               262656

 dense_43 (Dense)            (None, 256)               131328

 dense_44 (Dense)            (None, 10)                2570

=================================================================
Total params: 921,354
Trainable params: 921,354
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 70.96212410926819
```

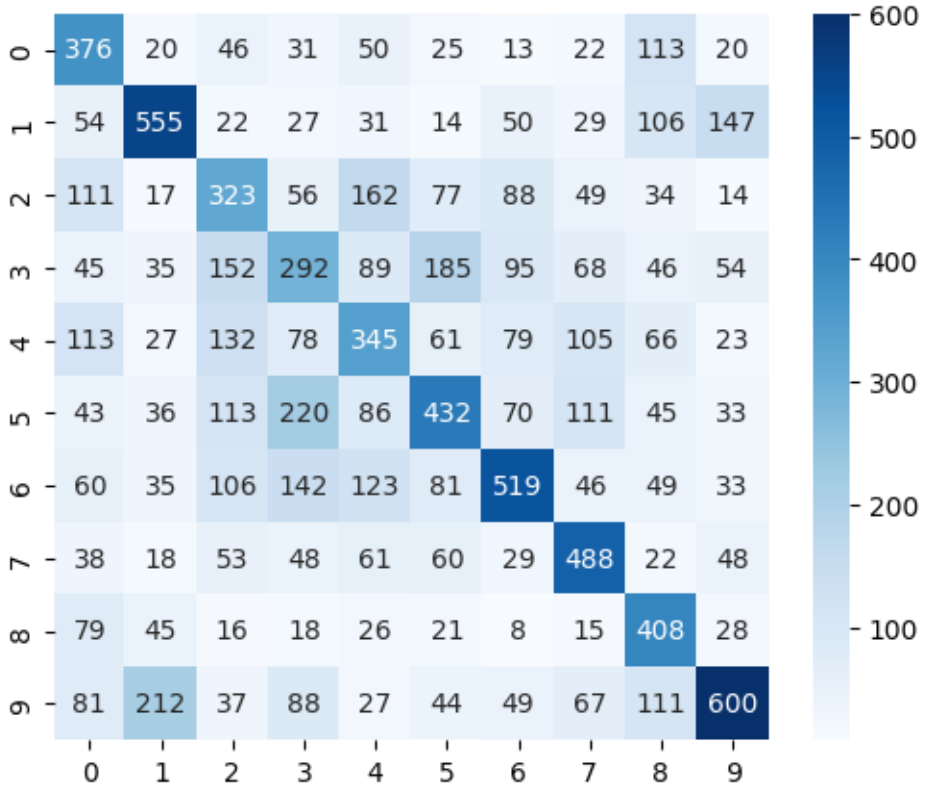
شکل ۱-۱۵ زمان آموزش مدل



شکل ۱-۱۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۸ ماتریس آشفتگی

```
accuracy on test data: 0.43380001187324524
loss on test data: 1.6441357135772705
f1_score on test data: 0.43413042463058754
recall on test data: 0.4338
presicion on test data: 0.44513499857923644
```

شکل ۱-۱۹ معیار های مختلف روی داده تست

با مقایسه نتایج بالا می بینیم که افزایش تعداد نورون های شبکه باعث افزایش زمان آموزش می شود، همچنین بهترین تعداد نورون برای لایه های شبکه همان مدل اول دارای دو لایه تماما متصل با ۱۰۲۴ نورون و یک لایه تماما متصل با ۵۱۲ نورون می باشد، در این حالت نسبت به دو حالت دیگر مدل ما دقت بیشتری دارد.

د)

بهترین مدل مرحله قبل یعنی دو لایه تماما متصل با ۱۰۲۴ نورون و یک لایه تماما متصل با ۵۱۲ نورون انتخاب می کنیم. حال نقش اندازه بسته را بررسی می کنیم

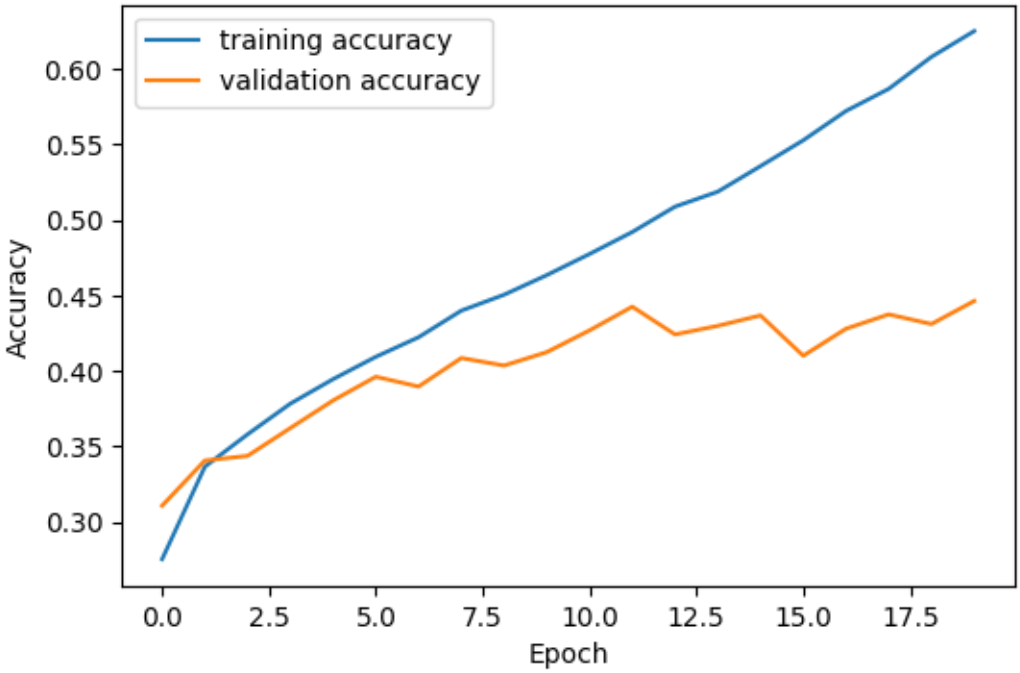
ابتدا سایز بسته را برابر ۳۲ قرار می دهیم

جدول ۱-۴ معماری شبکه برای قسمت د

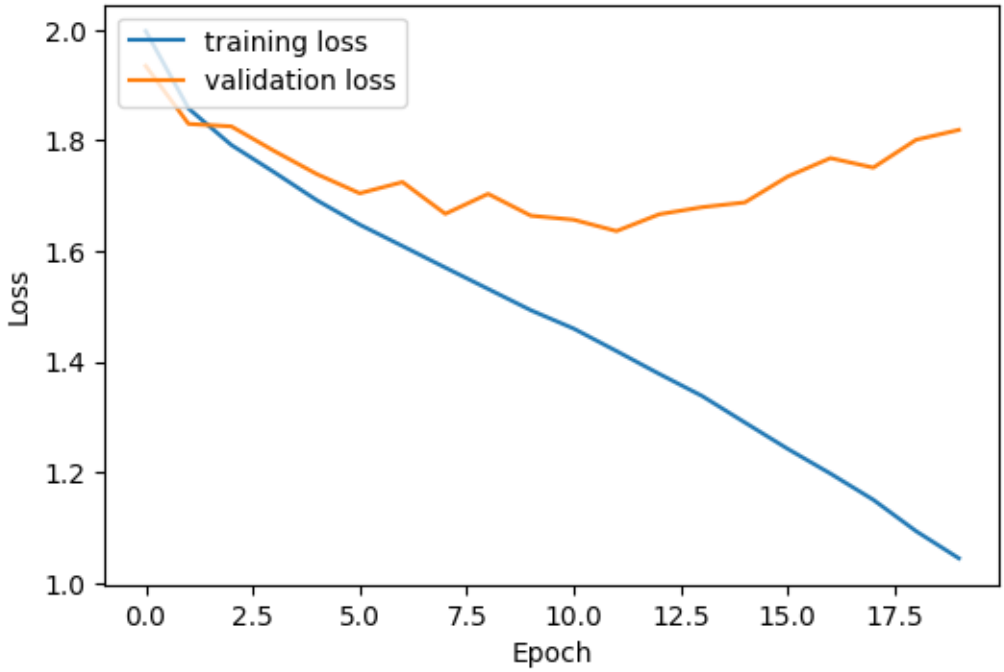
```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_49 (Dense)            (None, 1024)              1049600

 dense_50 (Dense)            (None, 1024)              1049600

 dense_51 (Dense)            (None, 512)               524800

 dense_52 (Dense)            (None, 10)                5130

=================================================================
Total params: 2,629,130
Trainable params: 2,629,130
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 202.48580193519592
```

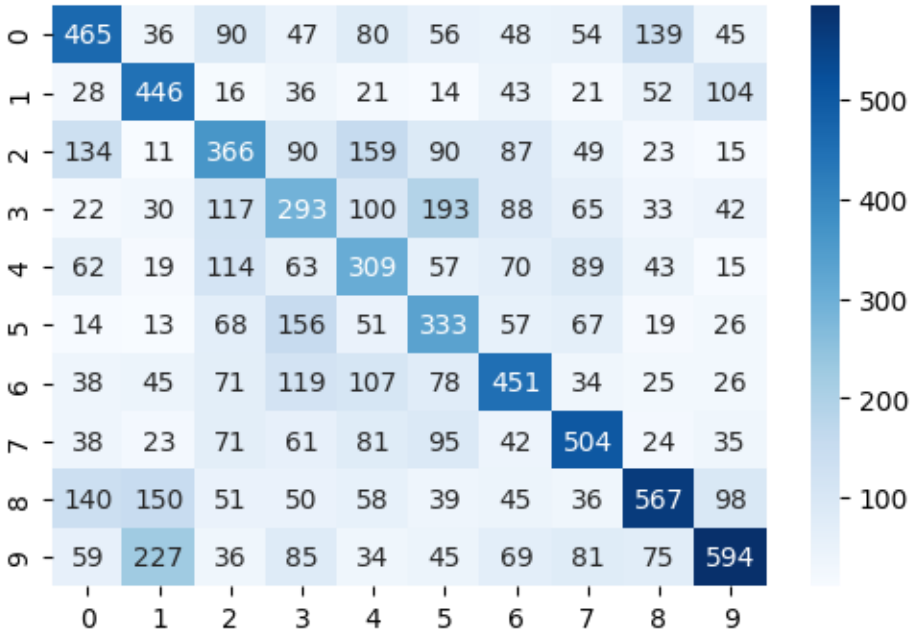
شکل ۱-۲۰ زمان آموزش



شکل ۱-۲۱ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۲۲ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل۱-۲۳ ماتریس آشفتگی

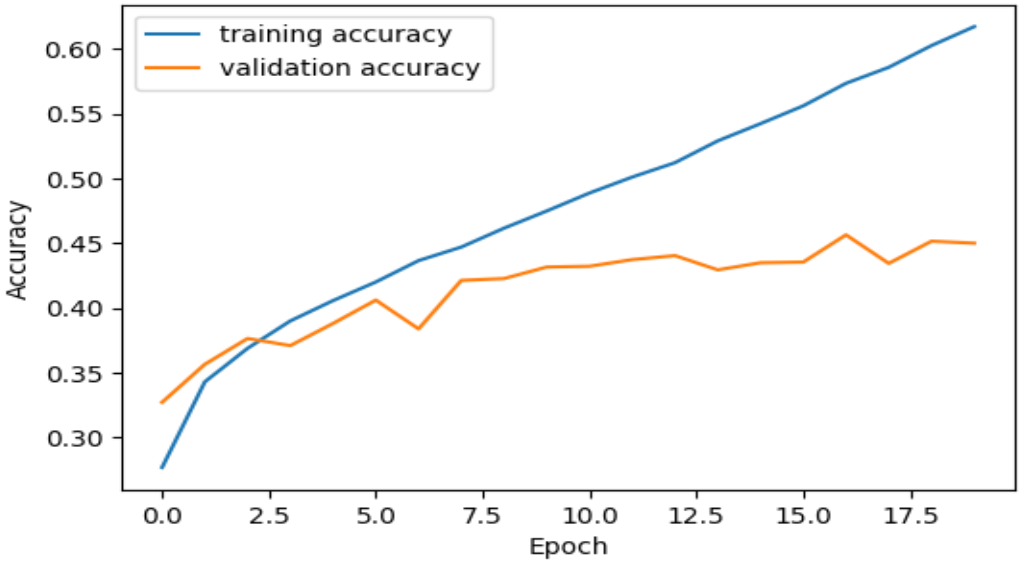
```
accuracy on test data: 0.4327999949455261
loss on test data: 1.8053287267684937
f1_score on test data: 0.43003435307407556
recall on test data: 0.4328
presicion on test data: 0.43326597526393307
```

شکل ۱-۲۴ معیار های مختلف برای داده تست

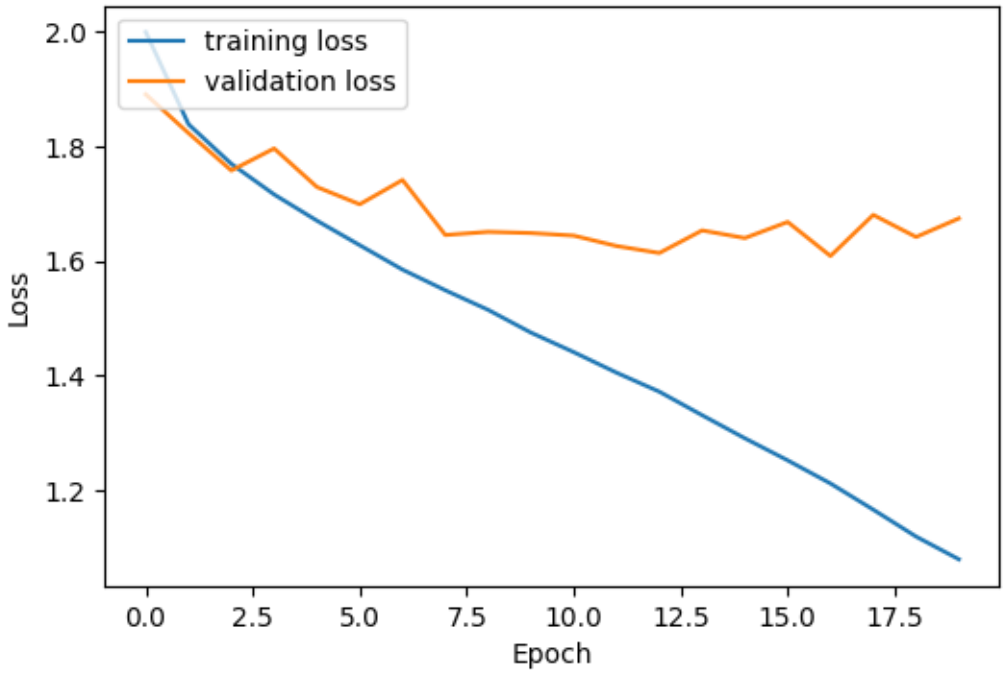
حال از سایز بسته ۶۴ استفاده می کنیم

```
training time: 86.24829077720642
```

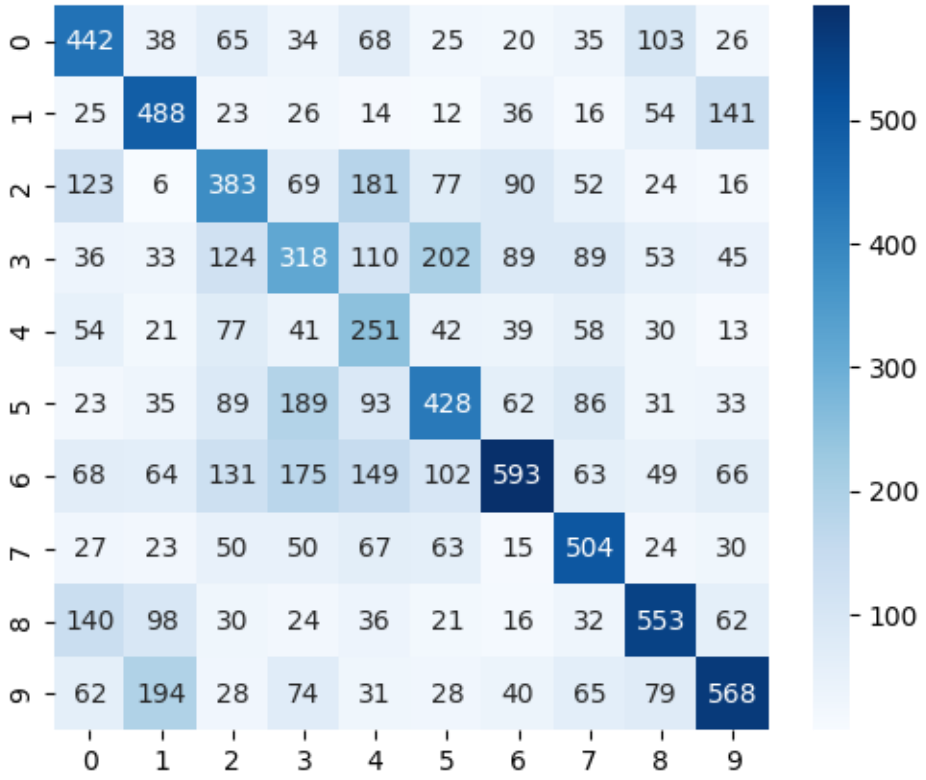
شکل ۱-۲۵ زمان آموزش



شکل ۱-۲۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۲۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۲۸ ماتریس آشفتگی

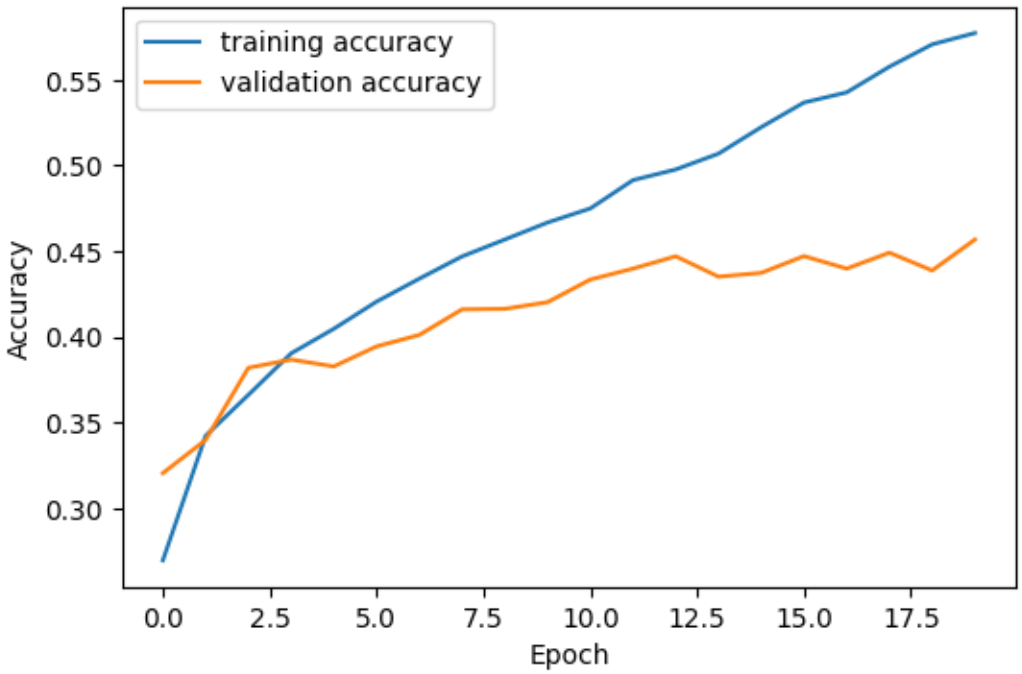
```
accuracy on test data: 0.4528000056743622
loss on test data: 1.6581695079803467
f1_score on test data: 0.4512196455685049
recall on test data: 0.4528
presicion on test data: 0.45959435594477716
```

شکل ۱-۲۹ معیار های مختلف برای داده های تست

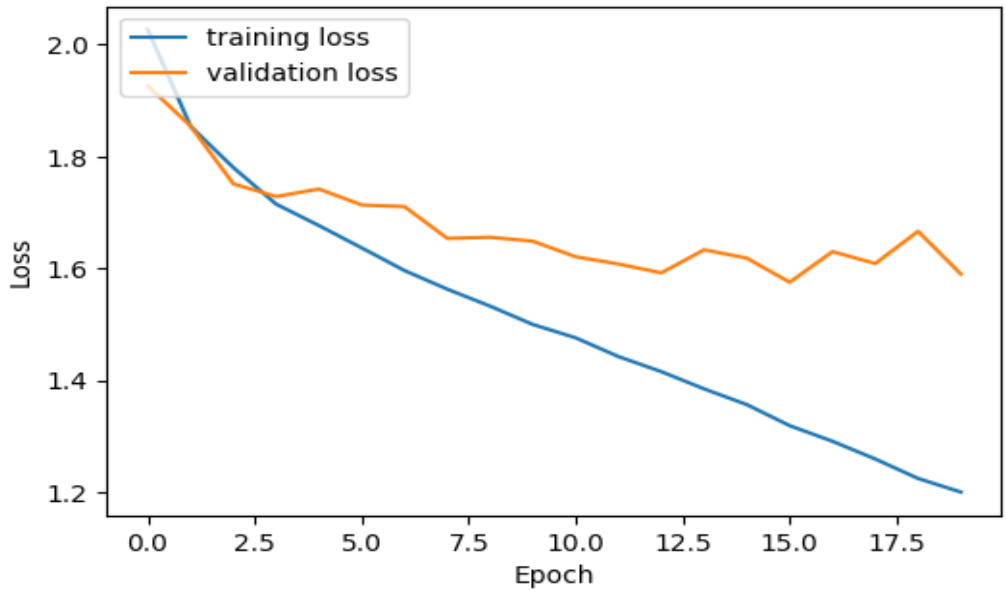
حال سایز بسته را برابر ۱۲۸ قرار می دهیم

```
training time: 51.29392194747925
```

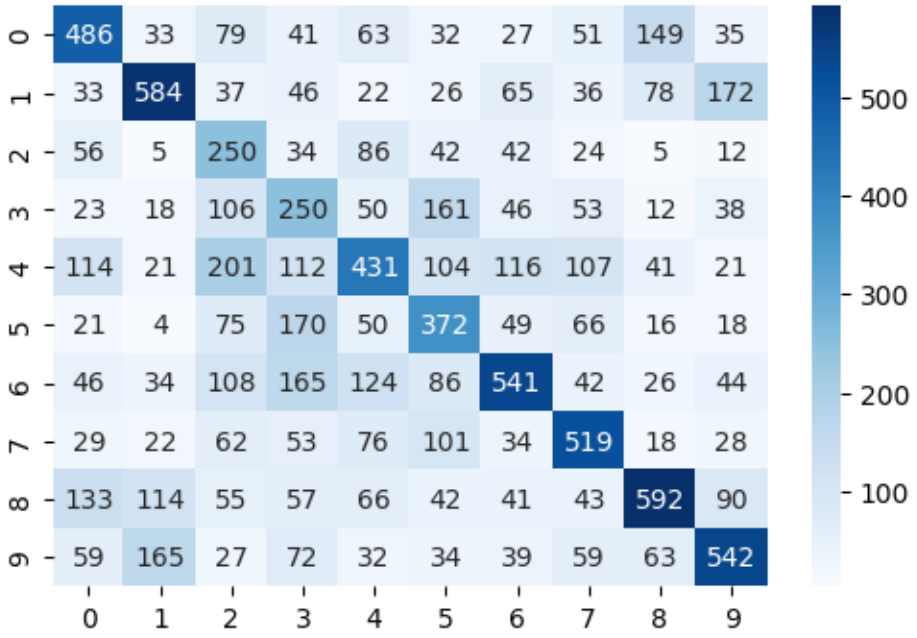
شکل ۱-۳۰ زمان آموزش



شکل ۱-۳۱ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۳۲ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۳۳ ماتریس آشفتگی

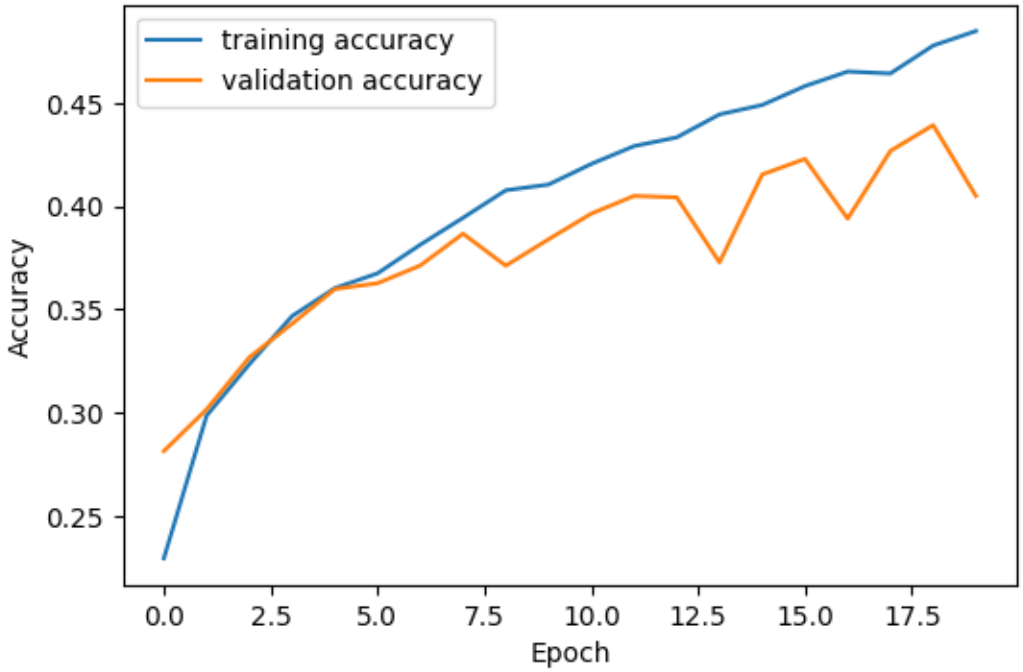
```
accuracy on test data: 0.45669999718666077
loss on test data: 1.569263219833374
f1_score on test data: 0.4504701374715907
recall on test data: 0.4567
presicion on test data: 0.45537946730309825
```

شکل ۱-۳۴ معیار های مختلف روی داده های تست

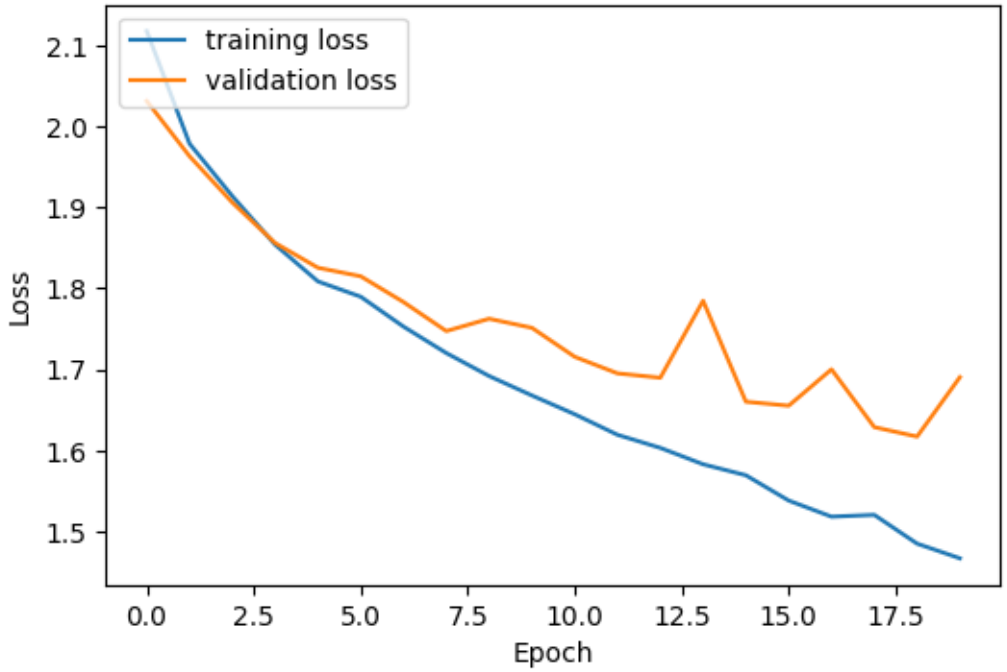
حال سایز بسته را برابر ۵۱۲ قرار می دهیم

```
training time: 42.225279808044434
```

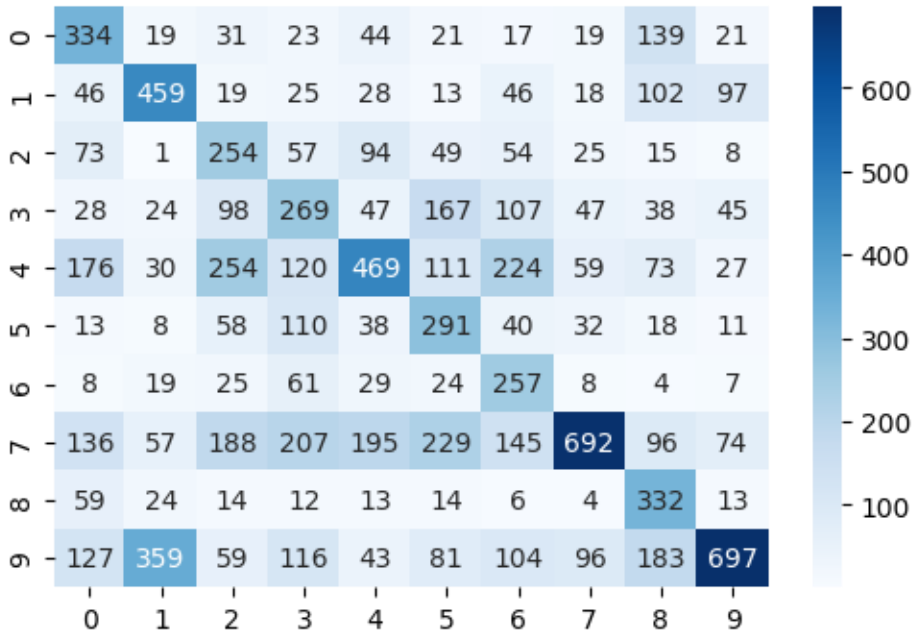
شکل ۱-۳۵ زمان آموزش



شکل ۱-۳۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۳۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۳۸ ماتریس آشفتگی

```
accuracy on test data: 0.405400007963180 54
loss on test data: 1.6742068529129028
f1_score on test data: 0.3970366253337889
recall on test data: 0.4054
presicion on test data: 0.4498626766807926
```

شکل ۱-۳۹ معیار های مختلف روی داده های تست

با توجه به نتایج بالا می بینیم که افزایش سایز بسته باعث کاهش زمان آموزش خواهد شد همچنین، افزایش سایز بسته تا ۱۲۸ باعث افزایش دقت و سایر معیار های مدل خواهد شد اما با افزایش سایز بسته به ۵۱۲ دقت مدل کاهش چشمگیری پیدا می کند و مدل خوب آموزش نمی بیند.

۵)

۱. تابع فعالساز ReLU

- مزایا: در مقابل کوچک شدن گرادیان[1] مقاوم است (به دلیل مشتق غیر صفر)، برای ورودی های مثبت یادگیری سریعتری ایجاد می کند.
- معایب: امکان انفجار گرادیان [2](بیش از حد بزرگ شدن به دلیل مشتق ثابت)، برای ورودی های کمتر از صفر خروجی صفر خواهد شد که باعث کاهش یادگیری می شود، حول صفر مرکزیت ندارد

۲. تابع فعالساز TanH

- مزایا: مرکزیت حول صفر، مشتق پذیری نرم در همه نقاط
- معایب: در مقابل کوچک شدن گرادیان مقاوم نیست، از نظر محاسباتی هزینه بر است

۳. تابع فعالساز Sigmoid

- مزایا: مشتق پذیری نرم در همه نقاط، امکان استفاده برای تعیین احتمال تعلق باینری به دلیل خروجی بین صفر و یک مناسب است
- معایب: در مقابل کوچک شدن گرادیان مقاوم نیست، از نظر محاسباتی هزینه بر است، حول صفر مرکزیت ندارد

۴. تابع فعالساز Softmax

- مزایا: مانند Sigmoid است با این تفاوت که امکان تعیین احتمال تعلق یک داده به دسته های مختلف (غیر باینری) را دارد
- معایب: همانند Sigmoid

منبع:

https://www.linkedin.com/pulse/top-10-activation-functions-advantages-disadvantages-/dash

با توجه به قسمت قبلی، اندازه هر بسته را برابر ۱۲۸ در نظر می گیریم.

ابتدا در تمام لایه ها از تابع فعالساز ReLU استفاده می کنیم

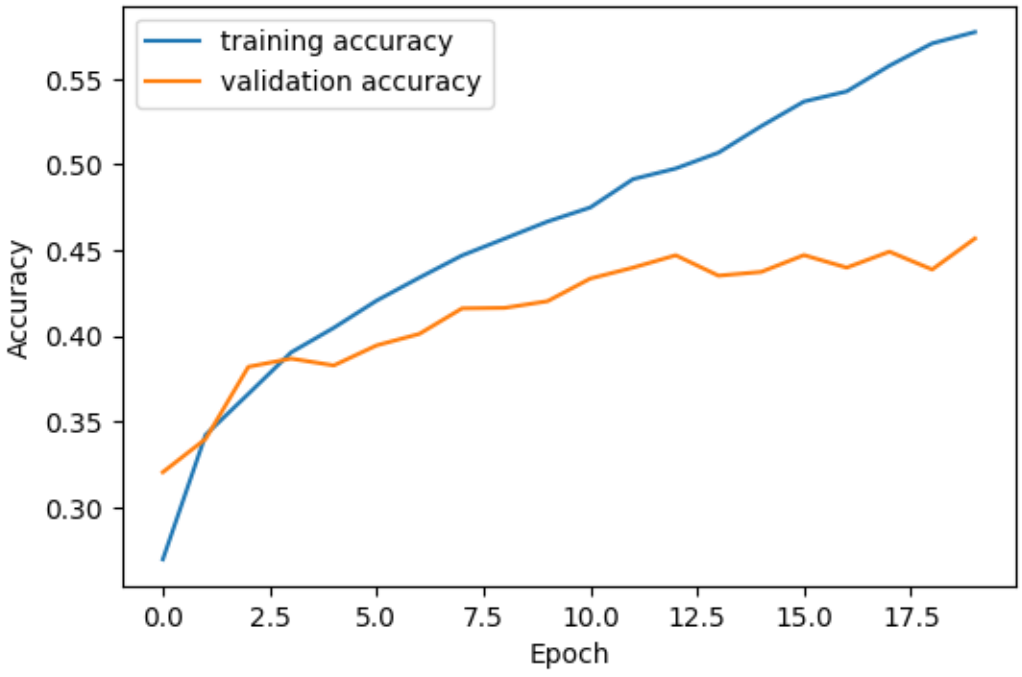
[1] Vanishing Gradient
[2] Gradient Explosion

جدول ۱-۴ معماری شبکه

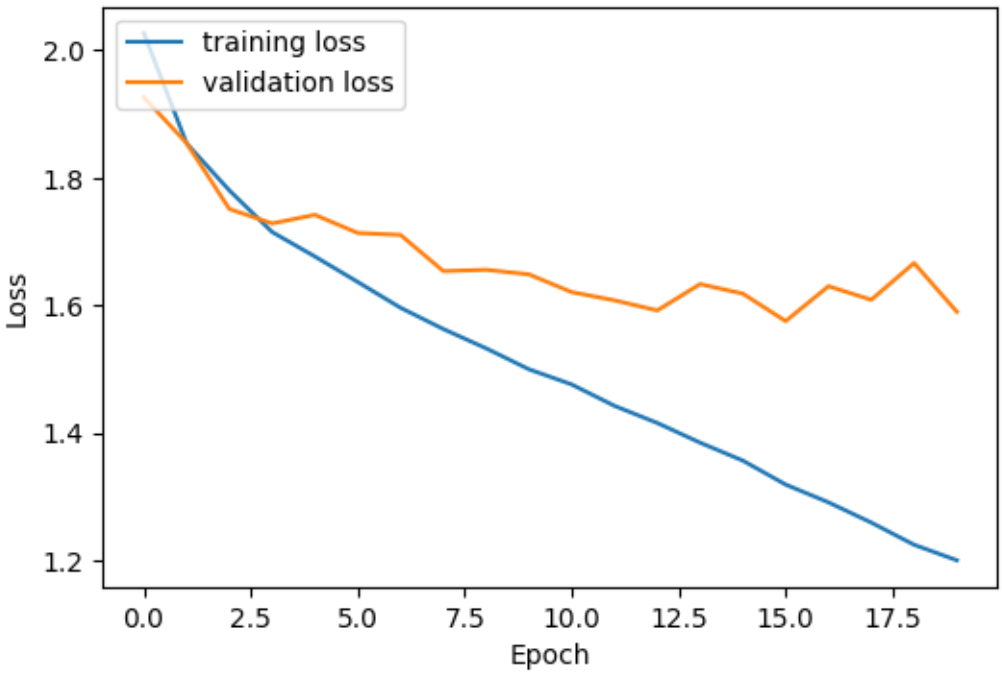
```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_69 (Dense)            (None, 1024)              1049600

 dense_70 (Dense)            (None, 1024)              1049600

 dense_71 (Dense)            (None, 512)               524800

 dense_72 (Dense)            (None, 10)                5130

=================================================================
Total params: 2,629,130
Trainable params: 2,629,130
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 51.29392194747925
```

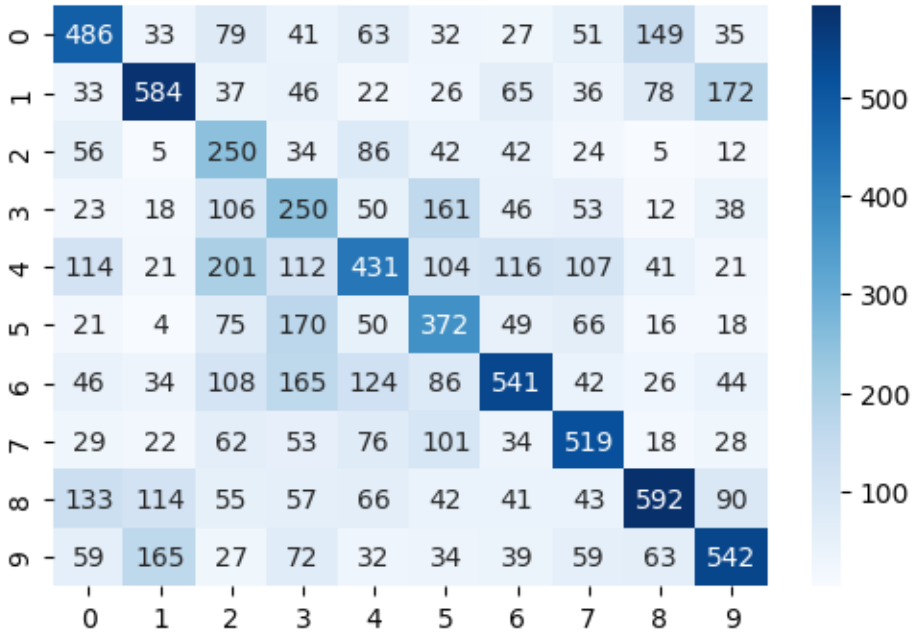
شکل ۱-۴۰ زمان آموزش



شکل ۱-۴۱ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۴۲ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۴۳ ماتریس آشفتگی

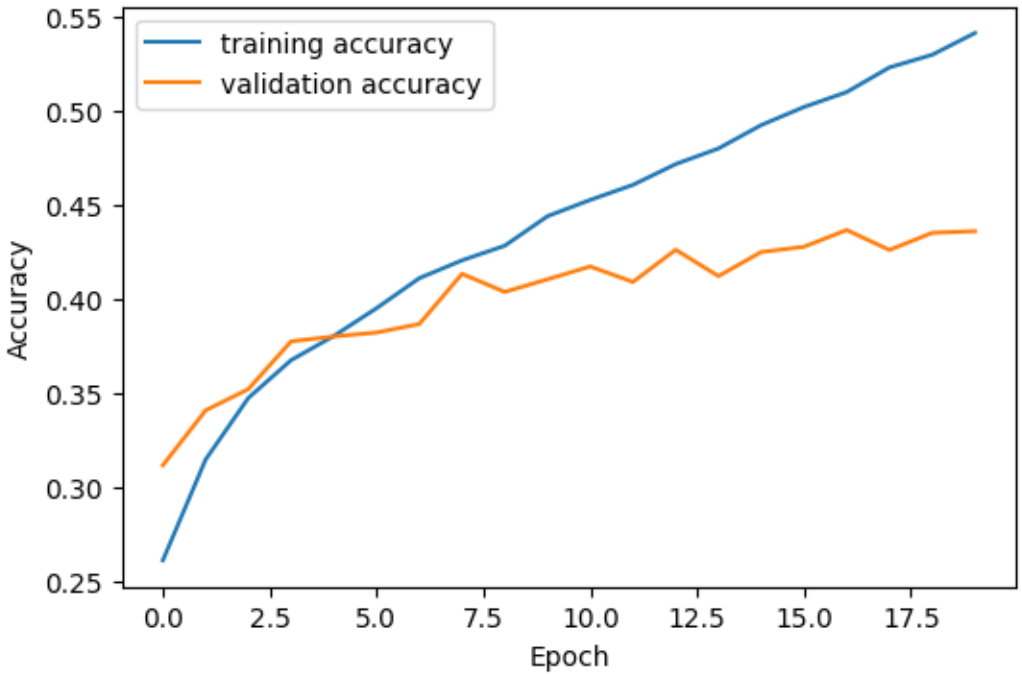
```
accuracy on test data: 0.45669999718666077
loss on test data: 1.569263219833374
f1_score on test data: 0.4504701374715907
recall on test data: 0.4567
presicion on test data: 0.45537946730309825
```

شکل ۱-۴۴ معیار های مختلف برای داده های تست

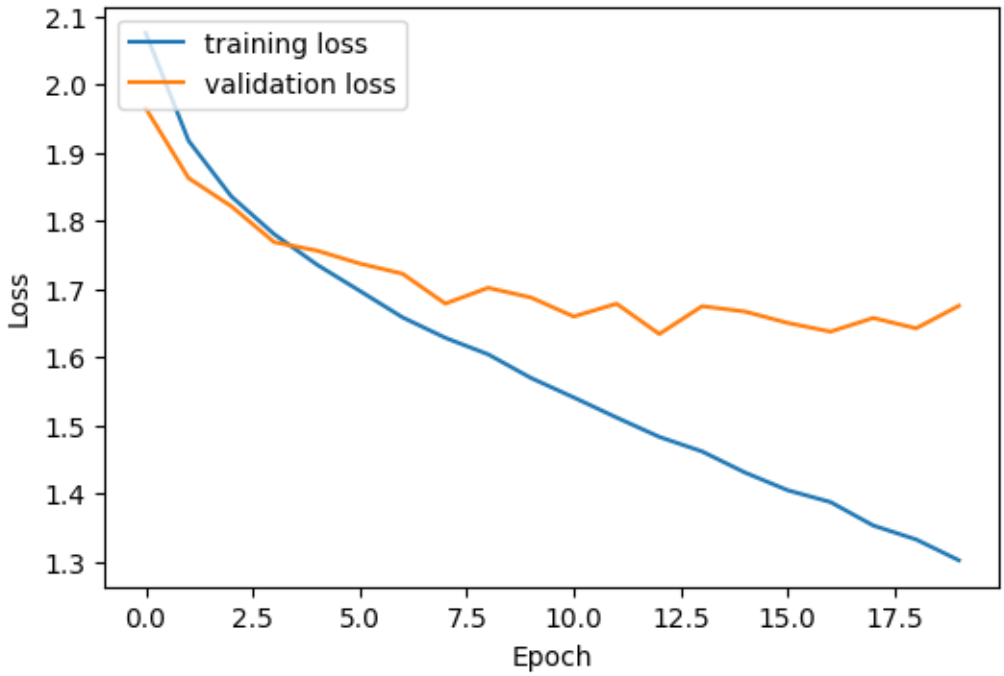
حال از تابع فعالساز Tanh استفاده می کنیم.

```
training time: 83.18342351913452
```

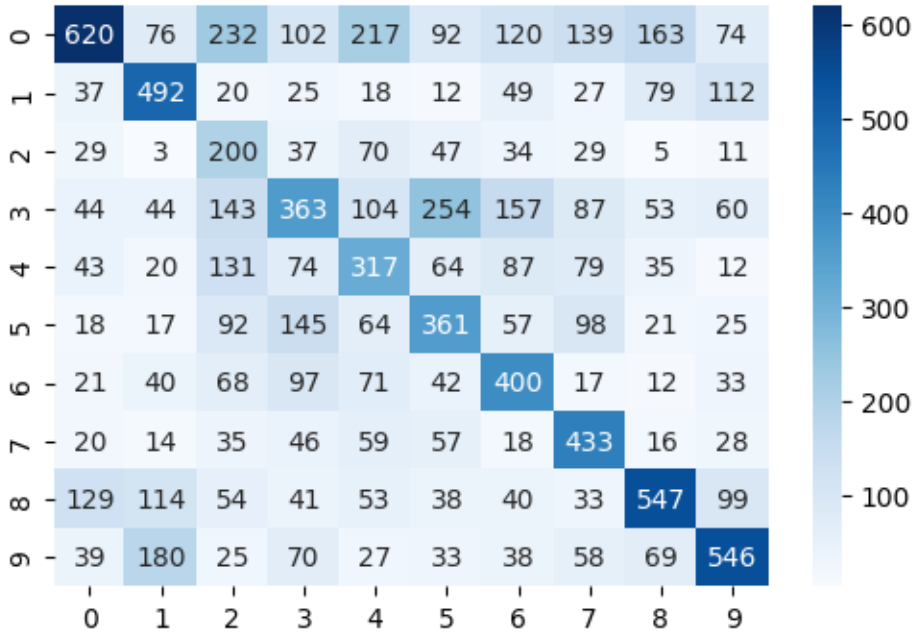
شکل ۱-۴۵ زمان آموزش



شکل ۱-۴۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۴۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۴۸ ماتریس آشفتگی

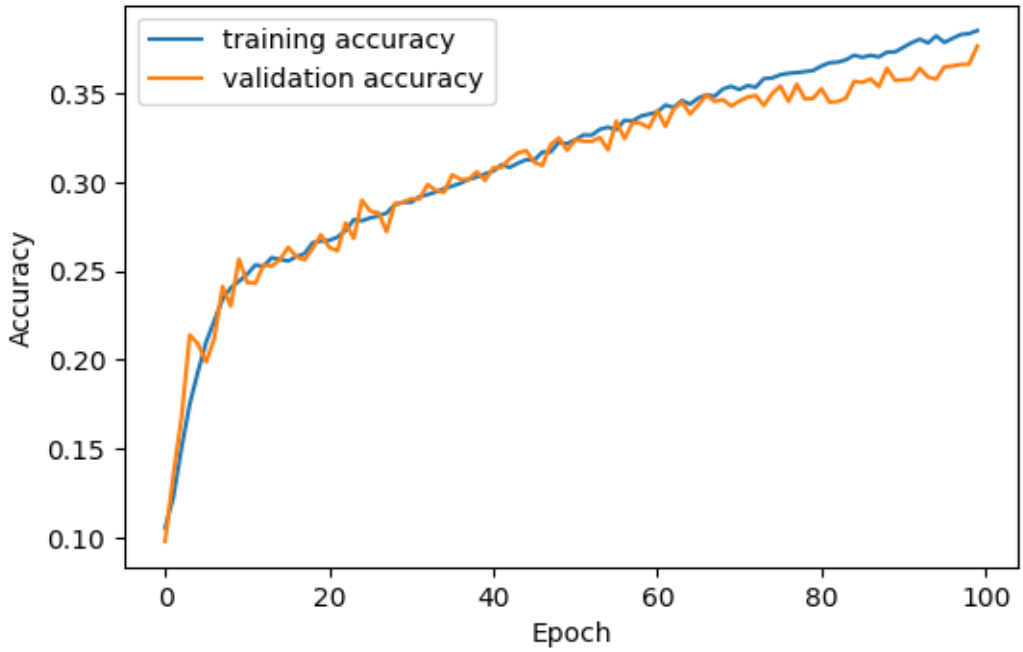
```
accuracy on test data: 0.4278999865055084
loss on test data: 1.666632056236267
f1_score on test data: 0.42506534066974255
recall on test data: 0.4279
presicion on test data: 0.4455416108547209
```

شکل ۱-۴۹ معیار های مختلف روی داده های تست

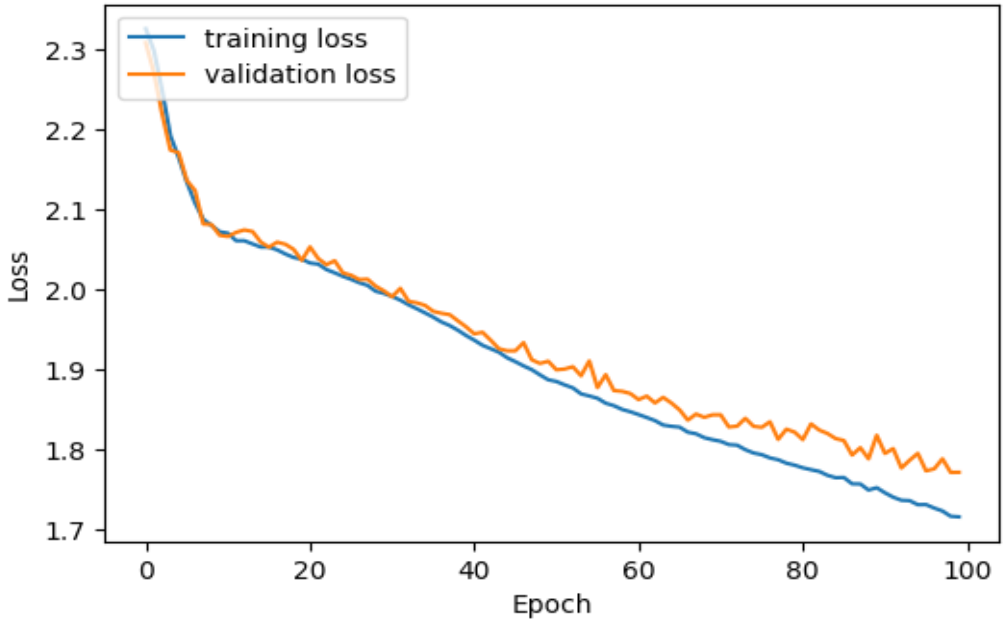
حال از تابع فعالساز Sigmoid در هر لایه استفاده می کنیم. به دلیل اینکه در ۲۰ تکرار خطای داده های ارزیابی پا به پای خطای داده های آموزش کاهش می یابد از ۱۰۰ تکرار برای این بخش استفاده می کنیم

```
training time: 263.9474573135376
```

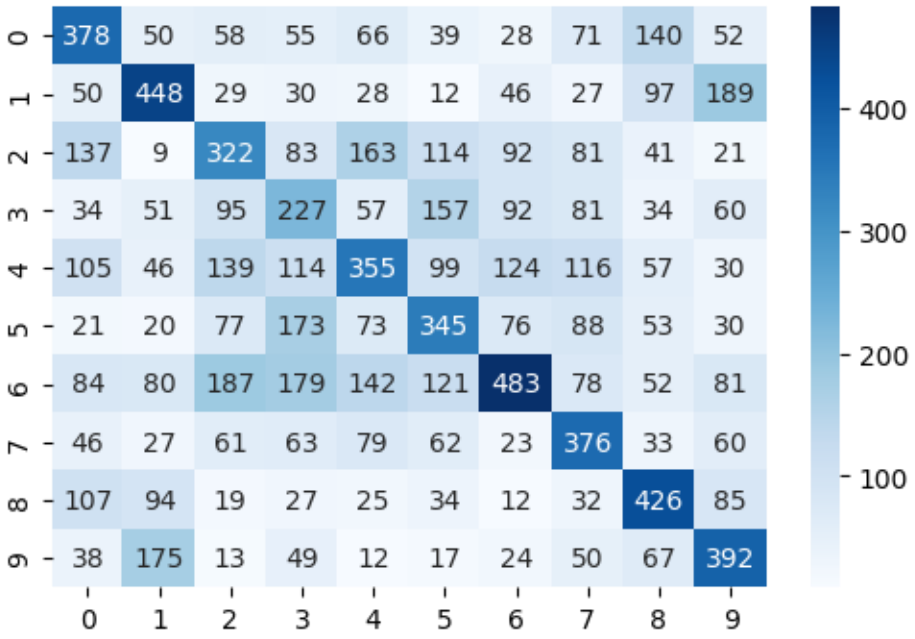
شکل ۱-۵۰ زمان آموزش



شکل ۱-۵۱ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۵۲ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۵۳ ماتریس آشفتگی

```
accuracy on test data: 0.3752000033855438
loss on test data: 1.76398766040802
f1_score on test data: 0.37626582760851585
recall on test data: 0.3752
presicion on test data: 0.38319779653881525
```

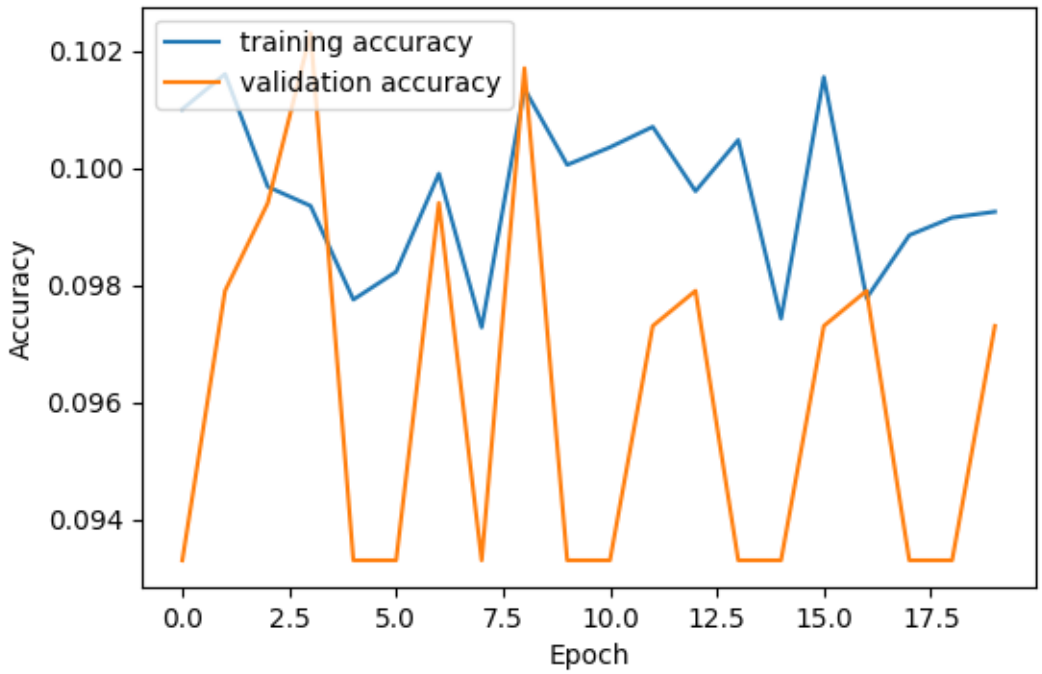
شکل ۱-۵۴ معیار های مختلف روی داده های تست

می بینیم با اینکه زمان آموزش بیشتر از مدل ها با تابع فعالساز دیگر است، اما مدل به دقت مناسبی دست نیافته است. زیرا همانطور که دیدیم به علت معایب گفته شده این تابع فعالساز یادگیری کندی دارد.

حال از تابع فعالساز Softmax در هر لایه استفاده می کنیم.

```
training time: 83.0312032699585
```

شکل ۱-۵۵ زمان آموزش



شکل ۱-۵۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی

شکل ۱-۵۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

```
accuracy on test data: 0.10000000149011612
loss on test data: 2.3026535511016846
f1_score on test data: 0.01818181818181818
recall on test data: 0.1
presicion on test data: 0.01
```

شکل ۱-۵۸ معیار های مختلف برای داده های تست

می بینیم که استفاده از تابع softmax در لایه های مخفی باعث می شود که مدل اصلا آموزش نبیند، این تابع فعالساز در لایه آخر برای تعیین احتمال تعلق کاربرد دارد.

تابع Tanh و Sigmoid به دلیل هزینه بر بودن محاسبات زمان بیشتری نسبت به تابع فعالساز ReLU می برند. و دقت آنها هم پایین تر می باشد. پس طبق نتایج بالا بهترین مدل استفاده از تابع فعالساز ReLU به همراه Softmax در لایه آخر می باشد.

و)

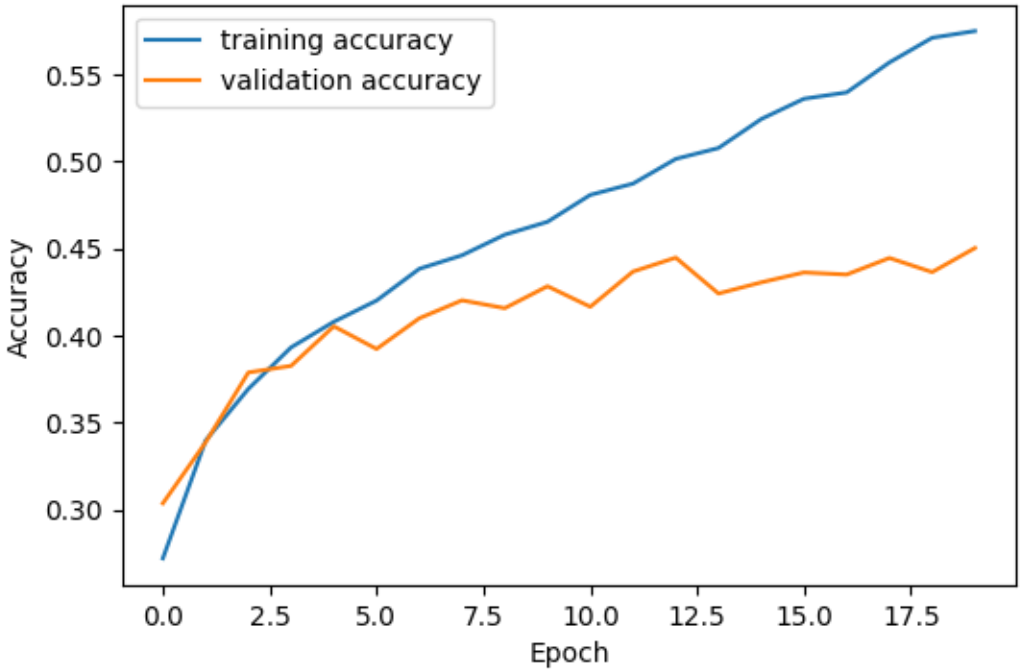
جدول ۱-۵ معماری شبکه

```
Model: "sequential_28"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_117 (Dense)           (None, 1024)              1049600

 dense_118 (Dense)           (None, 1024)              1049600

 dense_119 (Dense)           (None, 512)               524800

 dense_120 (Dense)           (None, 10)                5130

=================================================================
Total params: 2,629,130
Trainable params: 2,629,130
Non-trainable params: 0
_________________________________________________________________
```

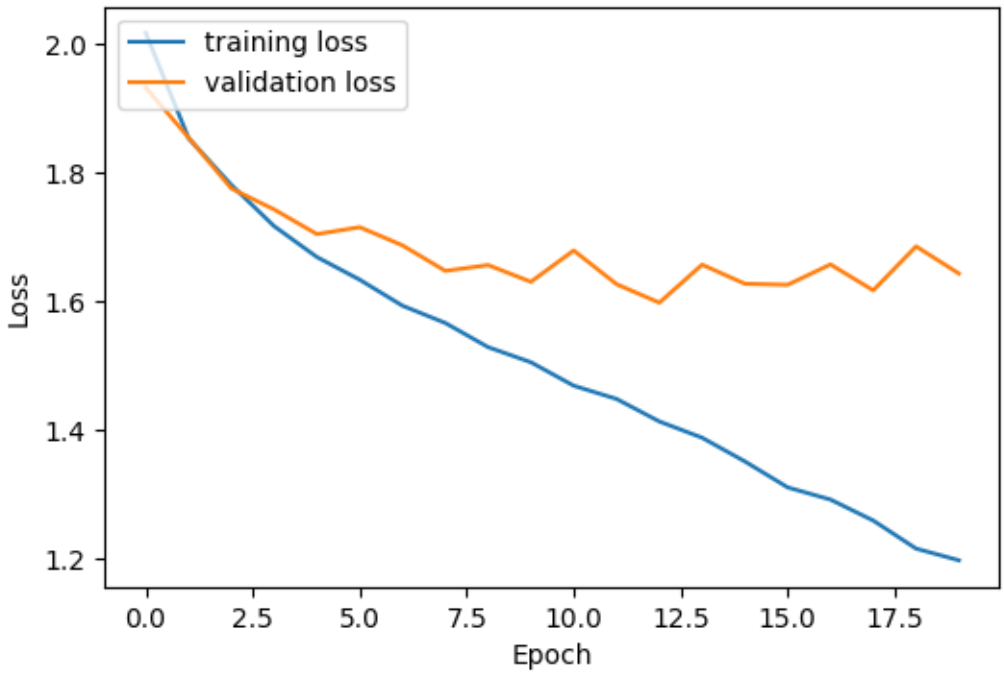
ابتدا از تابع خطای CategoricalCrossentropy استفاده می کنیم

```
training time: 51.277663707733154
```

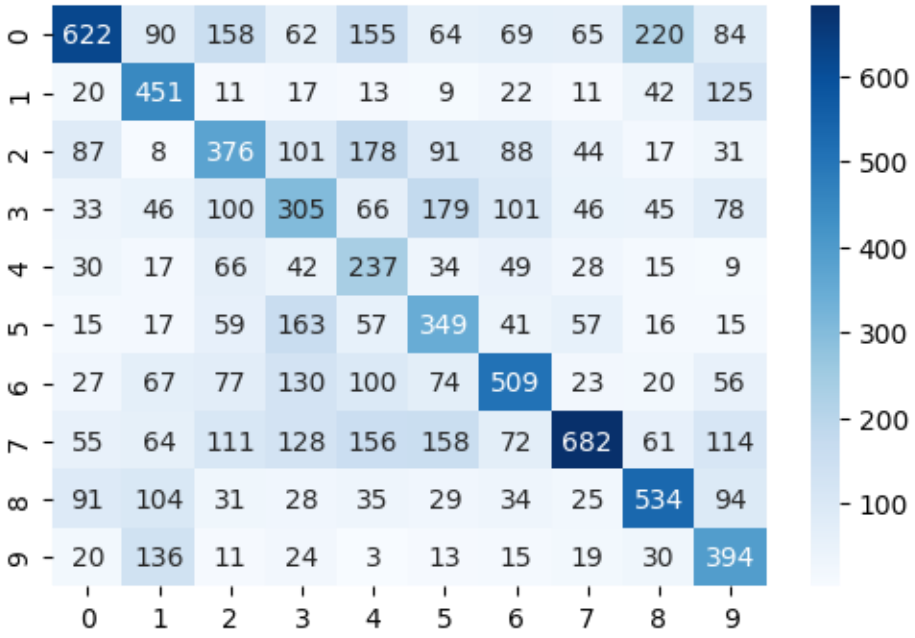
شکل ۱-۵۹ زمان آموزش



شکل ۱-۶۰ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۶۱ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۶۲ ماتریس آشفتگی

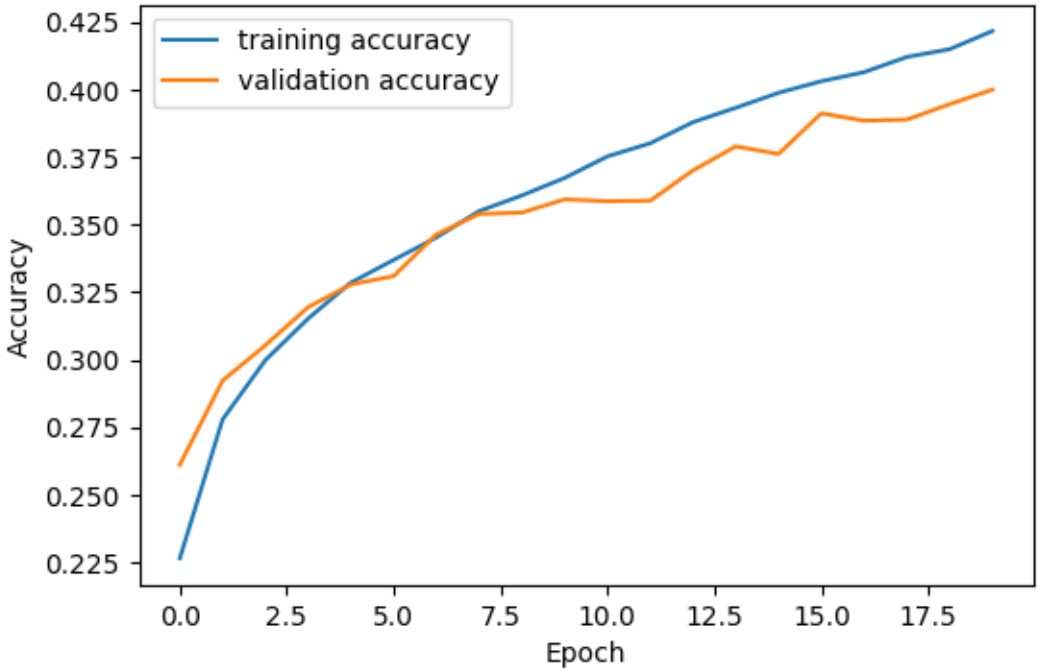
```
accuracy on test data: 0.44589999318122864
loss on test data: 1.632004737854004
f1_score on test data: 0.44015020460124427
recall on test data: 0.4459
presicion on test data: 0.46023794287624686
```

شکل ۱-۶۳ معیار های مختلف روی داده های تست

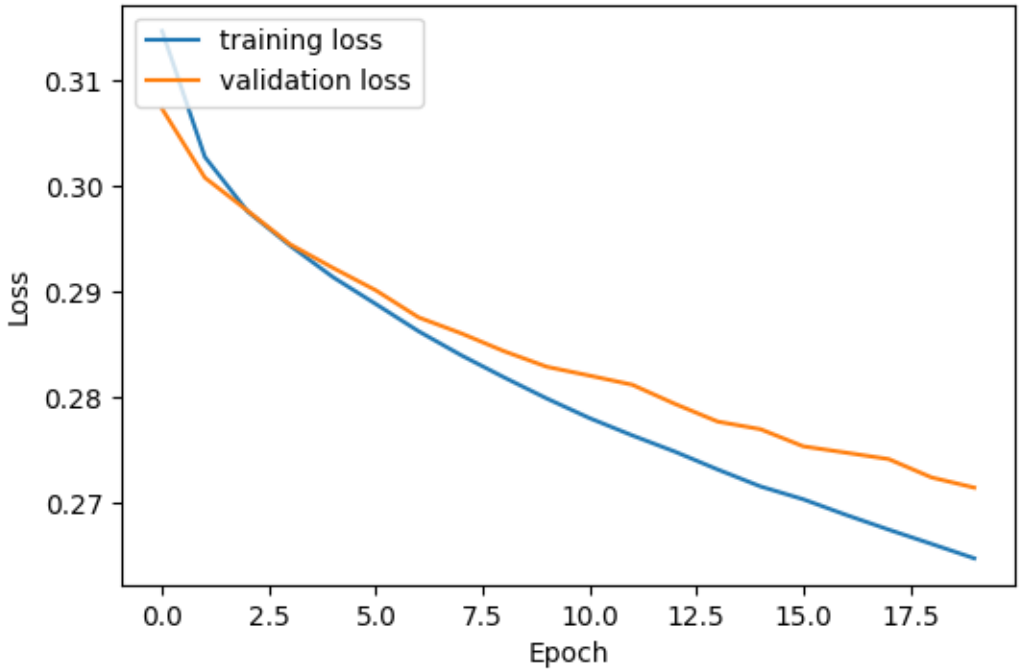
حال از تابع خطای Poisson استفاده می کنیم.

```
training time: 83.17326235771179
```

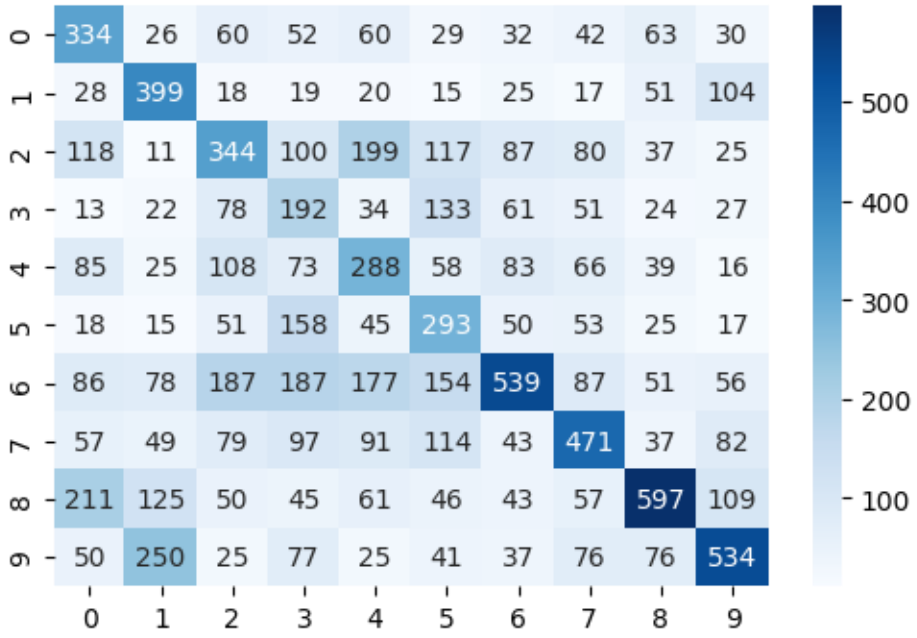
شکل ۱-۶۴ زمان آموزش



شکل ۱-۶۵ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۶۶ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۶۷ ماتریس آشفتگی

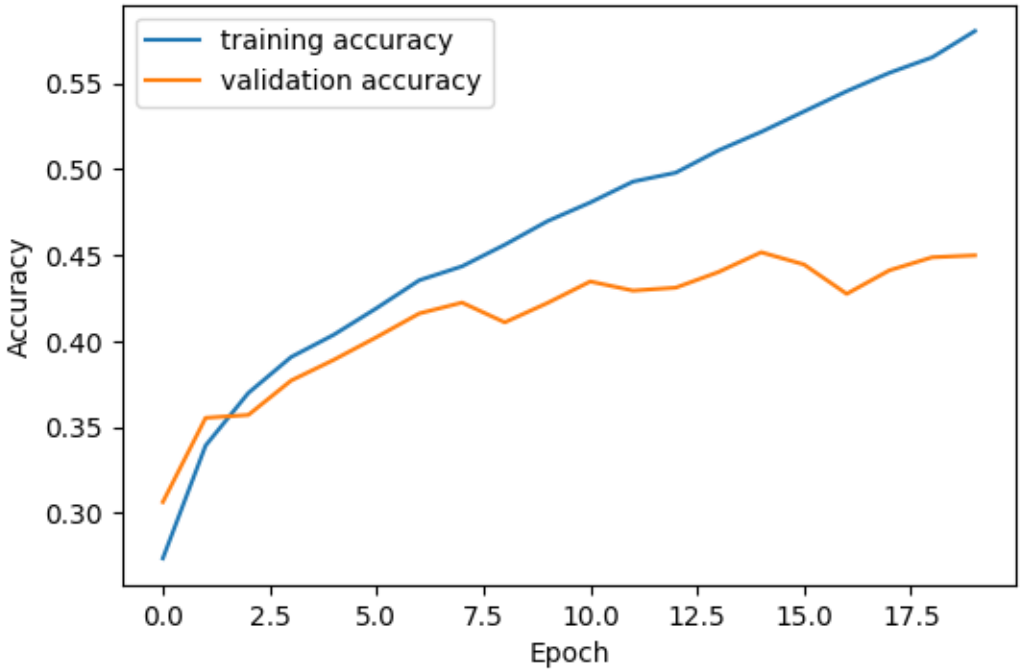
```
accuracy on test data: 0.39910000562667847
loss on test data: 0.269813597202301
f1_score on test data: 0.39248442969633646
recall on test data: 0.3991
presicion on test data: 0.40382582744543005
```

شکل ۱-۶۸ معیار های مختلف روی داده های تست

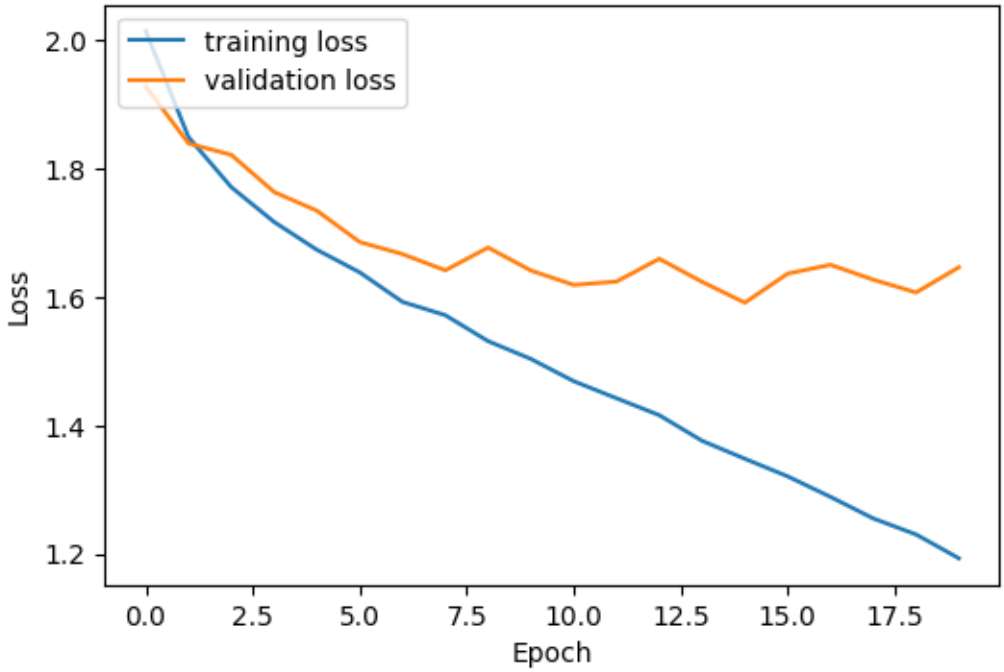
حال از تابع خطای Kullback Leibler Divergence Loss استفاده می کنیم.

```
training time: 83.05710649490356
```

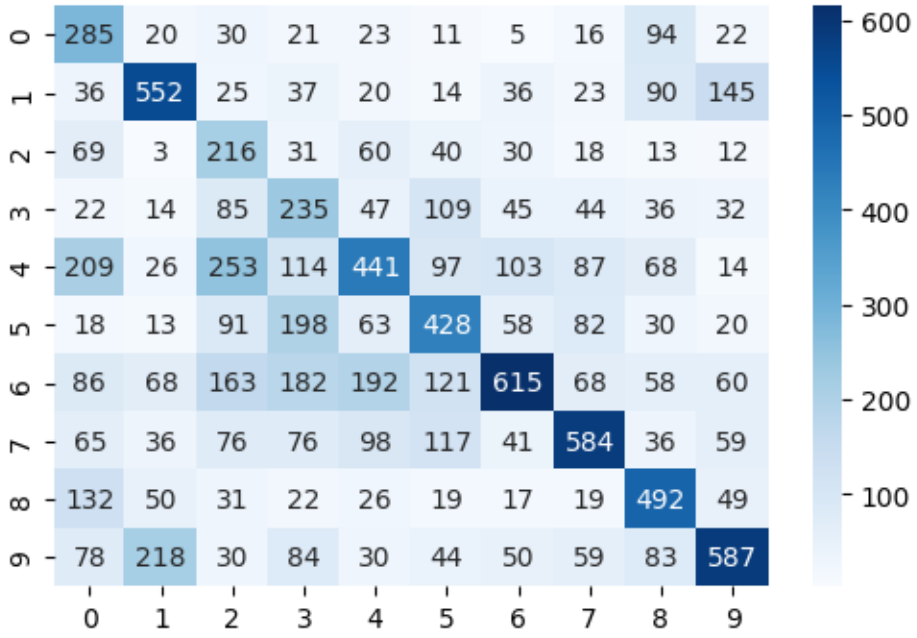
شکل ۱-۶۹ زمان آموزش



شکل ۱-۷۰ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۷۱ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۷۲ ماتریس آشفتگی

```
accuracy on test data: 0.443500012159347 53
loss on test data: 1.6471344232559204
f1_score on test data: 0.43492396564184244
recall on test data: 0.4435
presicion on test data: 0.4547126185623056
```

شکل ۱-۷۳ معیار های مختلف روی داده های تست

با توجه به نتایج بالا می بینیم که تابع خطا CategoricalCrossentropy زمان آموزش کمتری نسبت به دو تابع خطای دیگر نیاز دارد، تابع خطای Poisson دارای دقت کمتری نسبت به دو تابع خطای دیگر است، و دو تابع خطای Cross Entropy و Kullback Leibler Divergence Loss تقریبا دقت یکسانی دارند

ج)

در این سوال بهترین مدل قسمت قبل با تابع خطای CategoricalCrossentropy را استفاده می کنیم

ابتدا از روش بهینه سازی SGD با تکانه 0.9[1] استفاده می کنیم

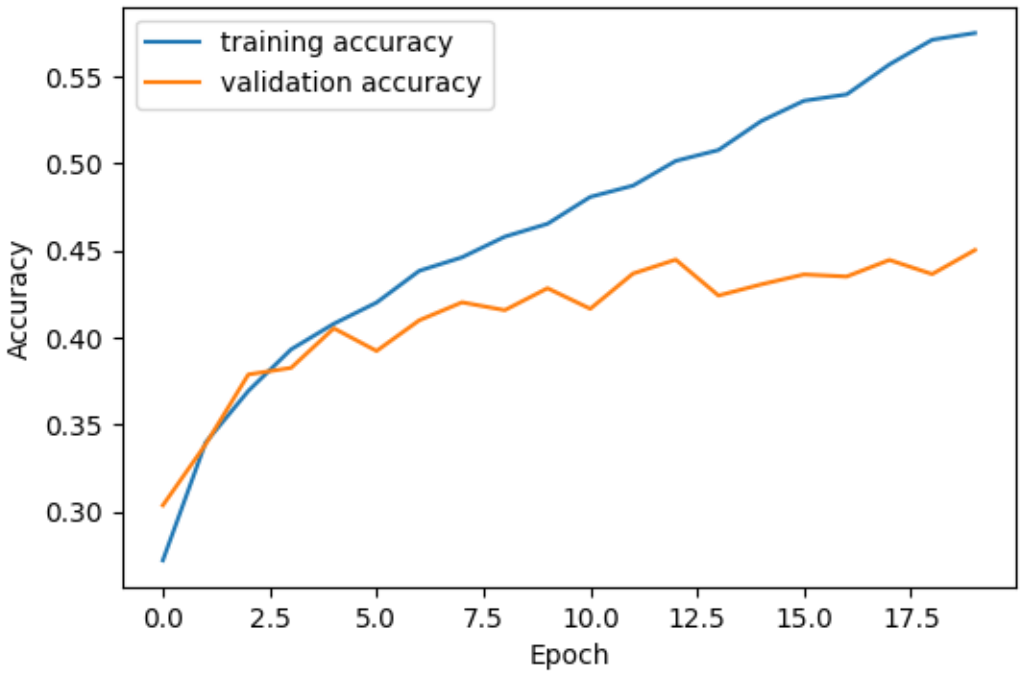
[1] Momentum

```
training time: 51.277663707733154
```

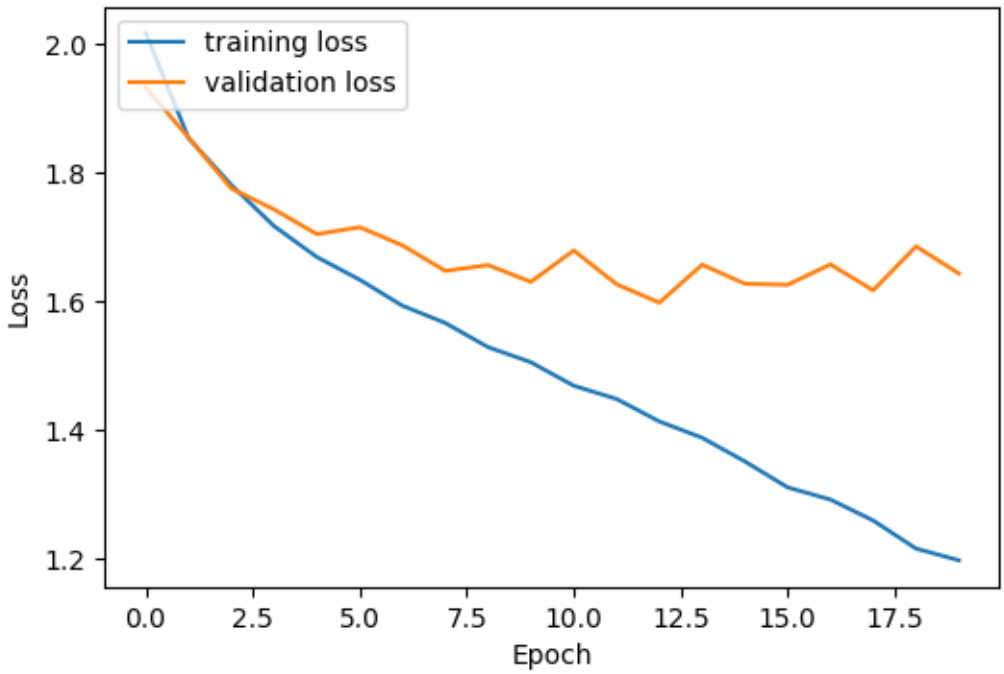
شکل ۱-۷۴ زمان آموزش

جدول ۱-۶ معماری شبکه

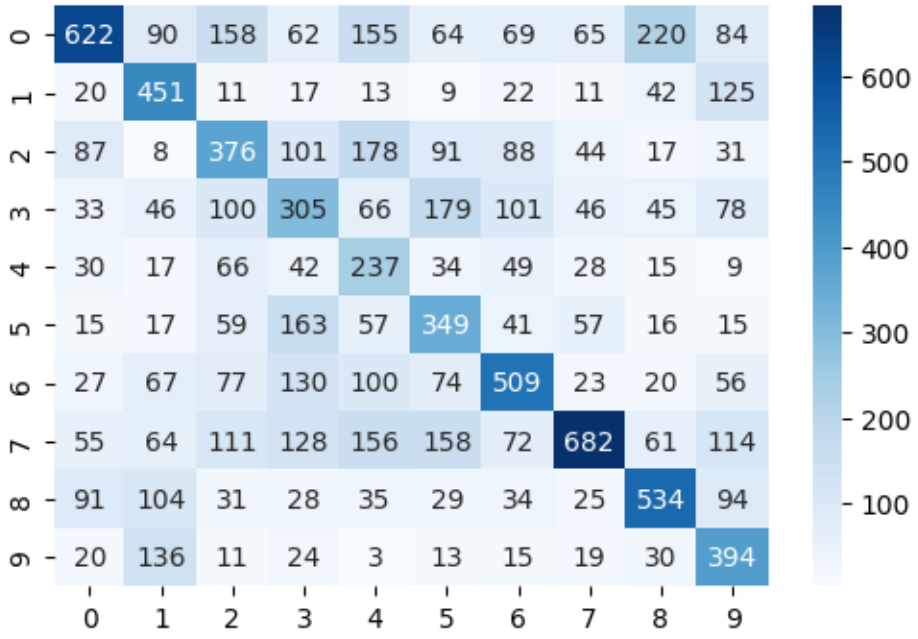
```
Model: "sequential_33"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_137 (Dense)            (None, 1024)              1049600

 dense_138 (Dense)            (None, 1024)              1049600

 dense_139 (Dense)            (None, 512)               524800

 dense_140 (Dense)            (None, 10)                5130

=================================================================
Total params: 2,629,130
Trainable params: 2,629,130
Non-trainable params: 0
_________________________________________________________________
```



شکل ۱-۷۵ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۷۶ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۷۷ ماتریس آشفتگی

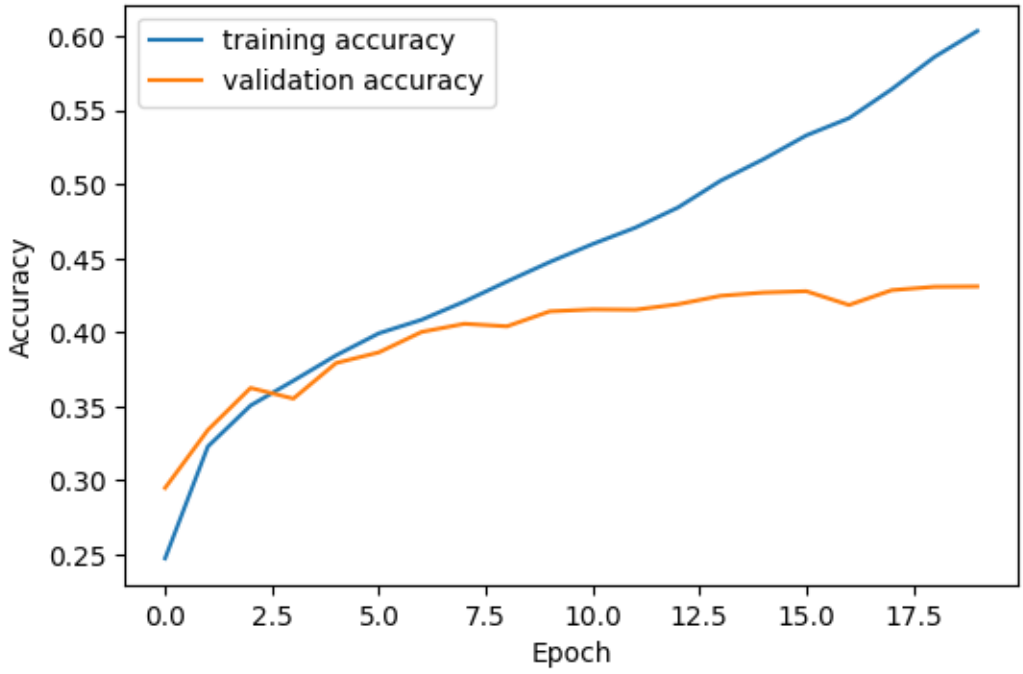
```
accuracy on test data: 0.44589999318122864
loss on test data: 1.632004737854004
f1_score on test data: 0.44015020460124427
recall on test data: 0.4459
presicion on test data: 0.46023794287624686
```

شکل ۱-۷۸ معیار های مختلف برای داده های تست

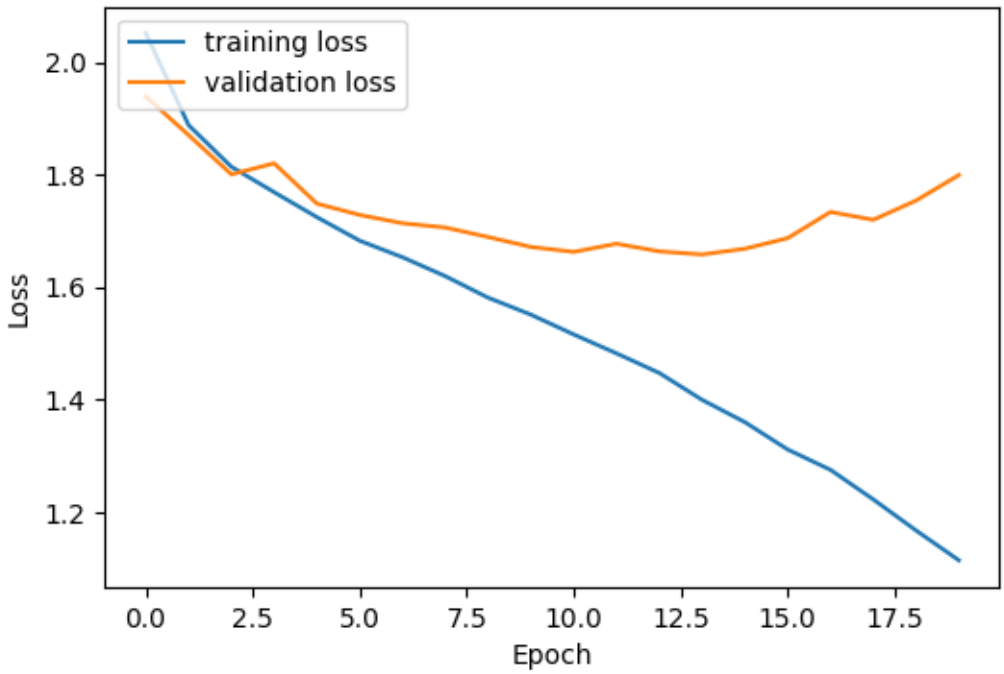
حال از روش بهینه سازی ADAM استفاده می کنیم

```
training time: 82.62482738494873
```

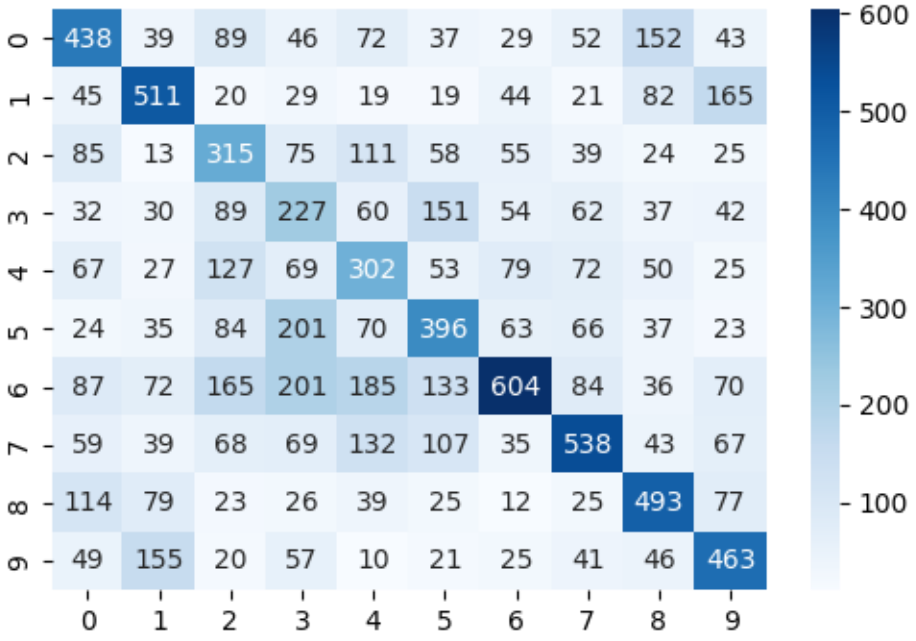
شکل ۱-۷۹ زمان آموزش



شکل ۱-۸۰ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۸۱ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۸۲ ماتریس آشفتگی

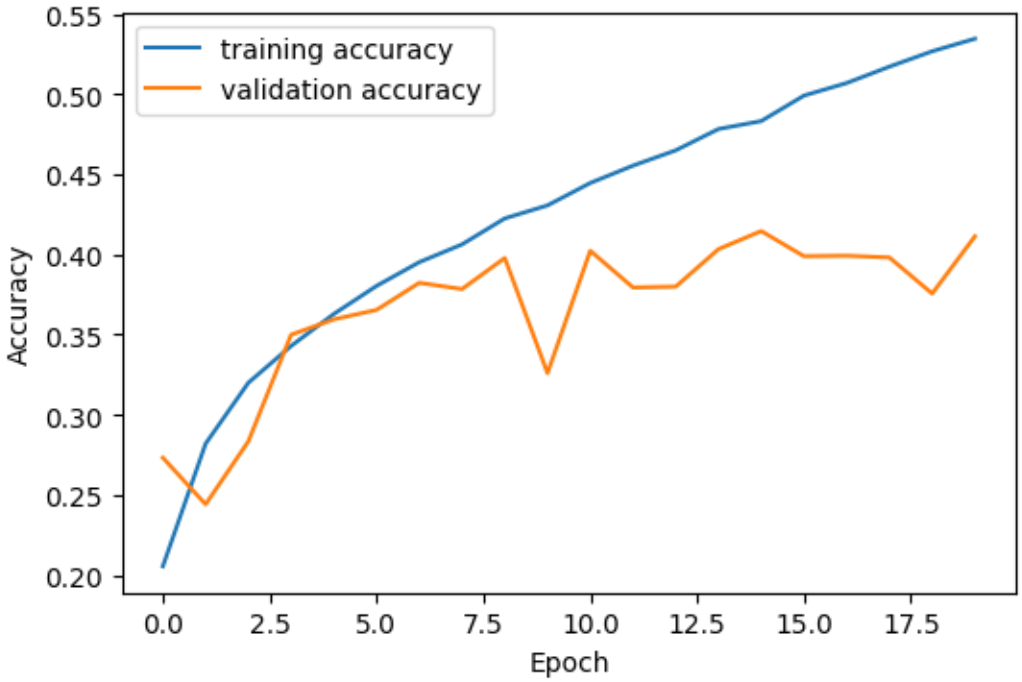
```
accuracy on test data: 0.4287000000476837
loss on test data: 1.7792353630065918
f1_score on test data: 0.42480086891904995
recall on test data: 0.4287
presicion on test data: 0.42967372124182873
```

شکل ۱-۸۳ معیار های مختلف برای داده های تست

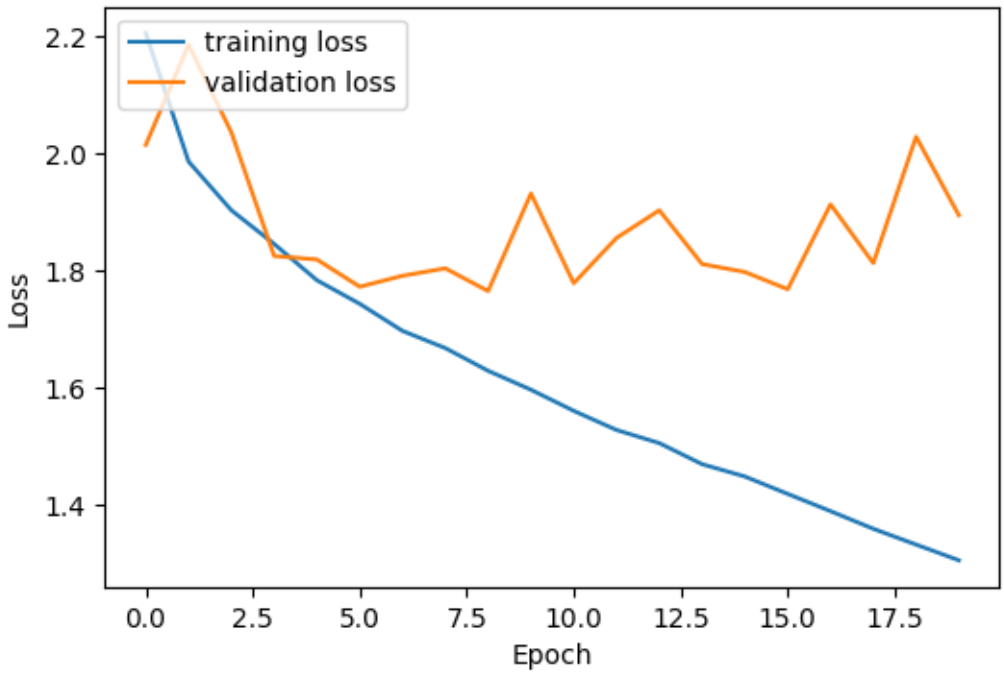
حال از روش بهینه سازی RMSprop استفاده می کنیم

```
training time: 83.59905242919922
```

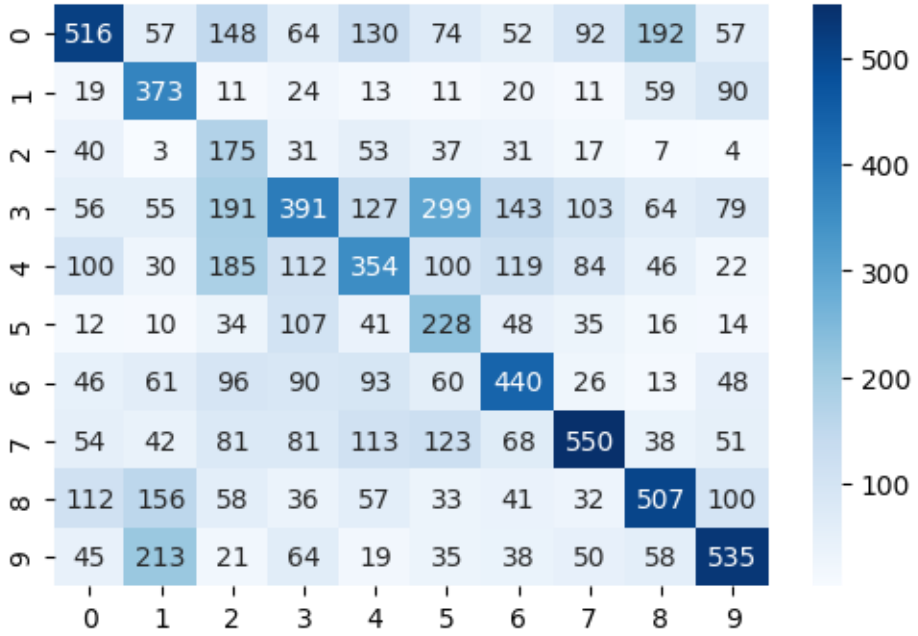
شکل ۱-۸۴ زمان آموزش



شکل ۱-۸۵ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۸۶ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۸۷ ماتریس آشفتگی

```
accuracy on test data: 0.40689998865127563
loss on test data: 1.8952937126159668
f1_score on test data: 0.4013261340552306
recall on test data: 0.4069
presicion on test data: 0.42434503767275966
```

شکل ۱-۸۸ معیار های مختلف برای داده های تست

با توجه به نتایج بالا می بینیم که بهینه ساز SGD با تکانه سریعتر از دو روش دیگر مدل را آموزش می دهد، همچنین دقت این روش در مقایسه با دو روش دیگر بهتر است.

SGD With Momentum > ADAM > RMSprop

ج)

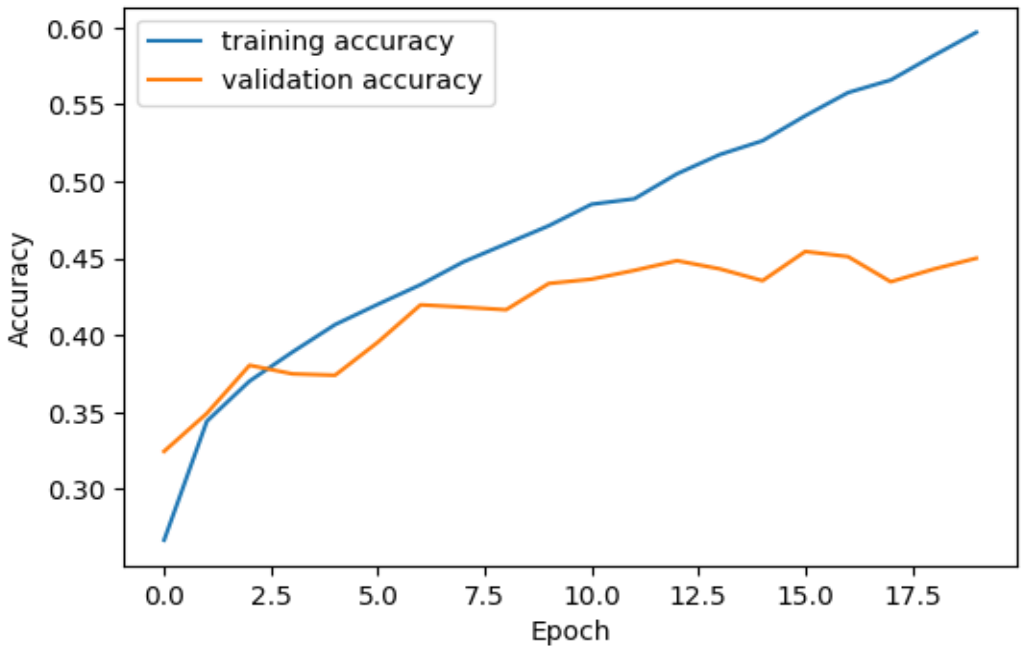
یک لایه با ۱۰۲۴ نورون به لایه های مخفی قسمت قبل اضافه می کنیم

جدول ۱-۷ معماری شبکه

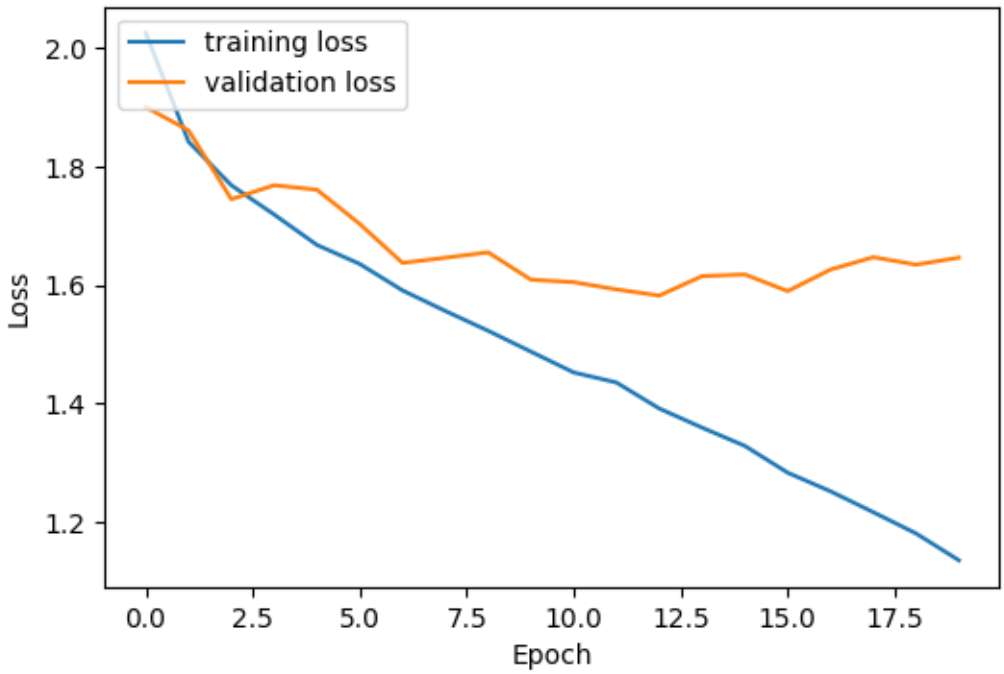
```
Model: "sequential_37"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_153 (Dense)            (None, 1024)              1049600

 dense_154 (Dense)            (None, 1024)              1049600

 dense_155 (Dense)            (None, 1024)              1049600

 dense_156 (Dense)            (None, 512)               524800

 dense_157 (Dense)            (None, 10)                5130

=================================================================
Total params: 3,678,730
Trainable params: 3,678,730
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 66.25007343292236
```

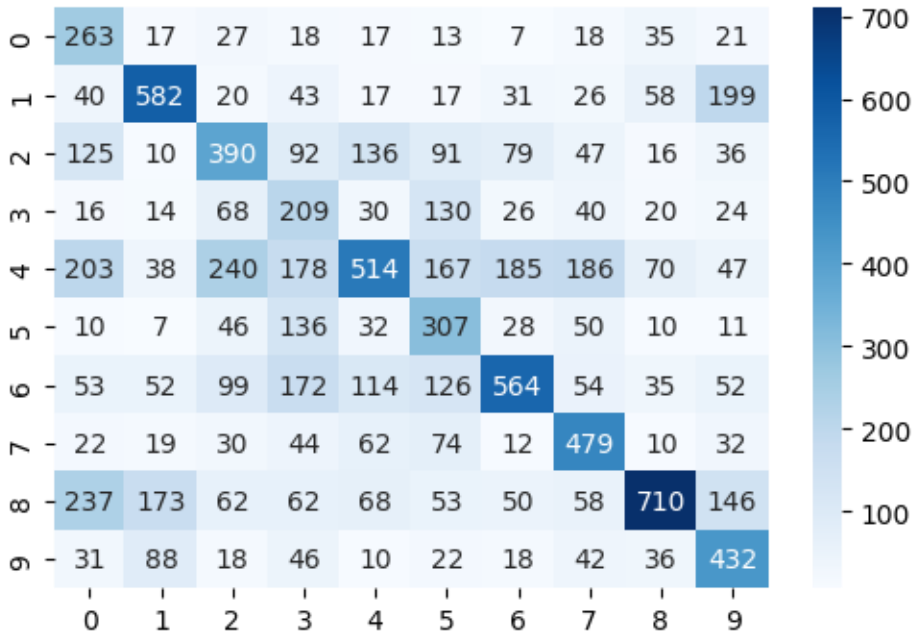
شکل ۱-۸۹ زمان آموزش



شکل ۱-۹۰ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۹۰ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۹۱ ماتریس آشفتگی

```
accuracy on test data: 0.4449999928474426
loss on test data: 1.6330419778823853
f1_score on test data: 0.43891308790346467
recall on test data: 0.445
presicion on test data: 0.4731457499006303
```

شکل ۱-۹۲ معیار های مختلف برای داده های تست

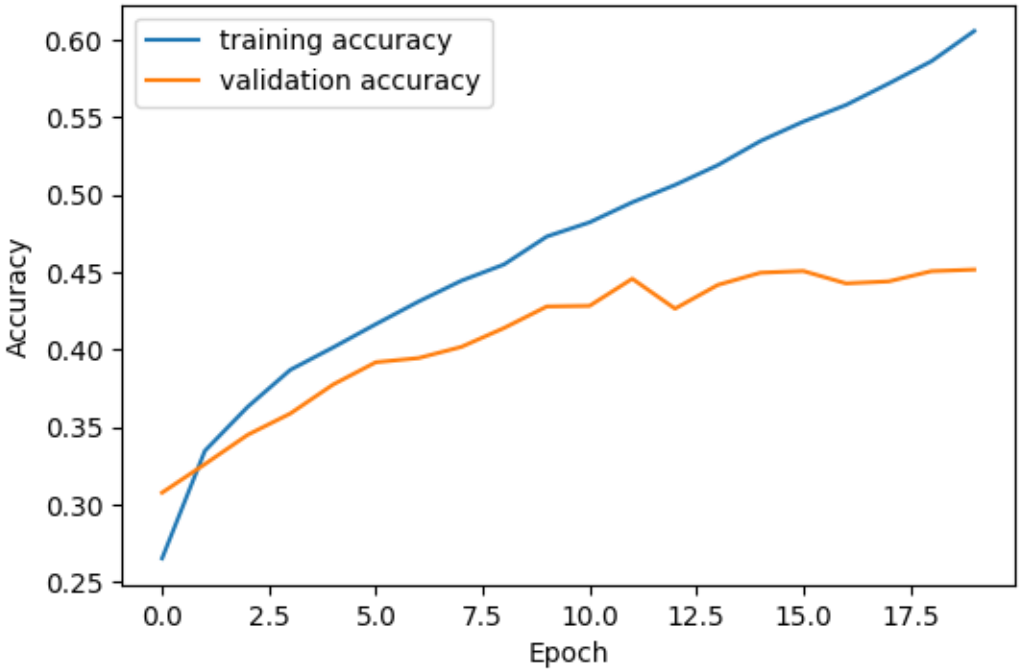
حال یک لایه مخفی دیگر با ۱۰۲۴ نورون اضافه می کنیم.

جدول ۱-۸ معماری شبکه

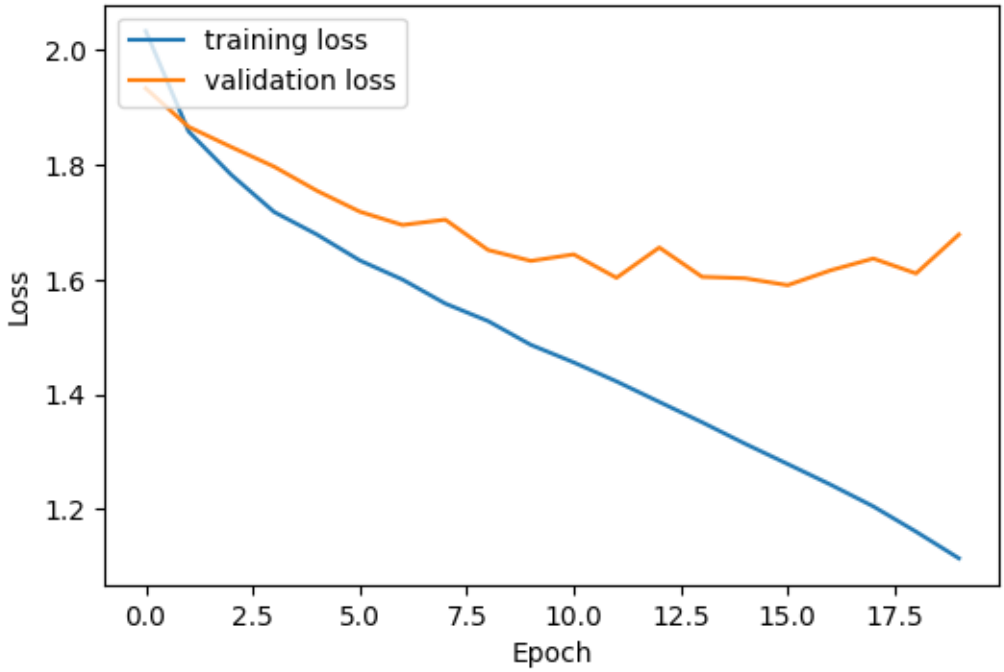
```
Model: "sequential_38"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_158 (Dense)           (None, 1024)              1049600

 dense_159 (Dense)           (None, 1024)              1049600

 dense_160 (Dense)           (None, 1024)              1049600

 dense_161 (Dense)           (None, 1024)              1049600

 dense_162 (Dense)           (None, 512)               524800

 dense_163 (Dense)           (None, 10)                5130

=================================================================
Total params: 4,728,330
Trainable params: 4,728,330
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 83.58302855491638
```

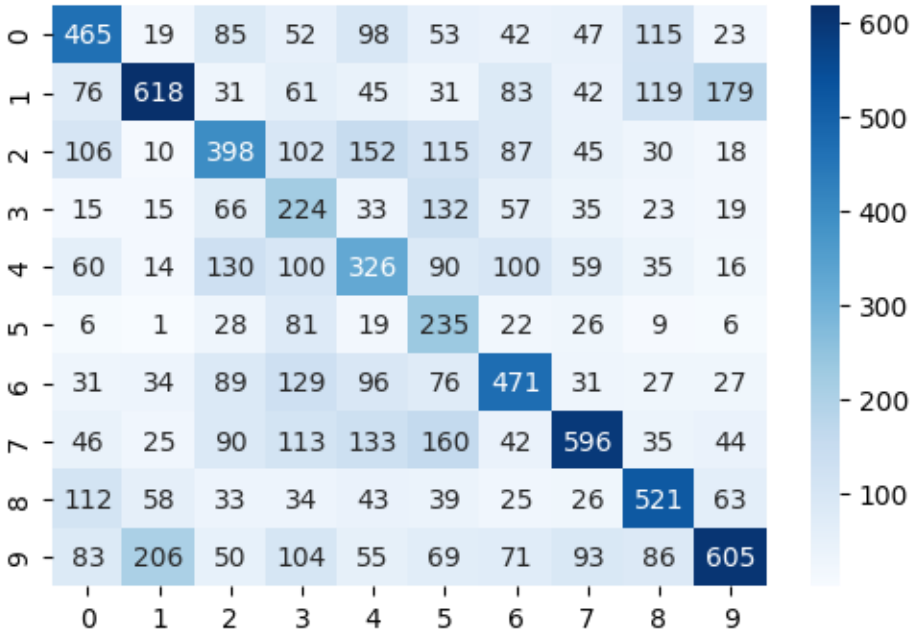
شکل ۱-۹۳ زمان آموزش



شکل ۱-۹۴ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۹۵ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۹۶ ماتریس آشفتگی

```
accuracy on test data: 0.44589999318122864
loss on test data: 1.6609435081481934
f1_score on test data: 0.4357685919874609
recall on test data: 0.4459
presicion on test data: 0.4477576678635779
```

شکل ۱-۹۷ معیار های مختلف برای داده های تست

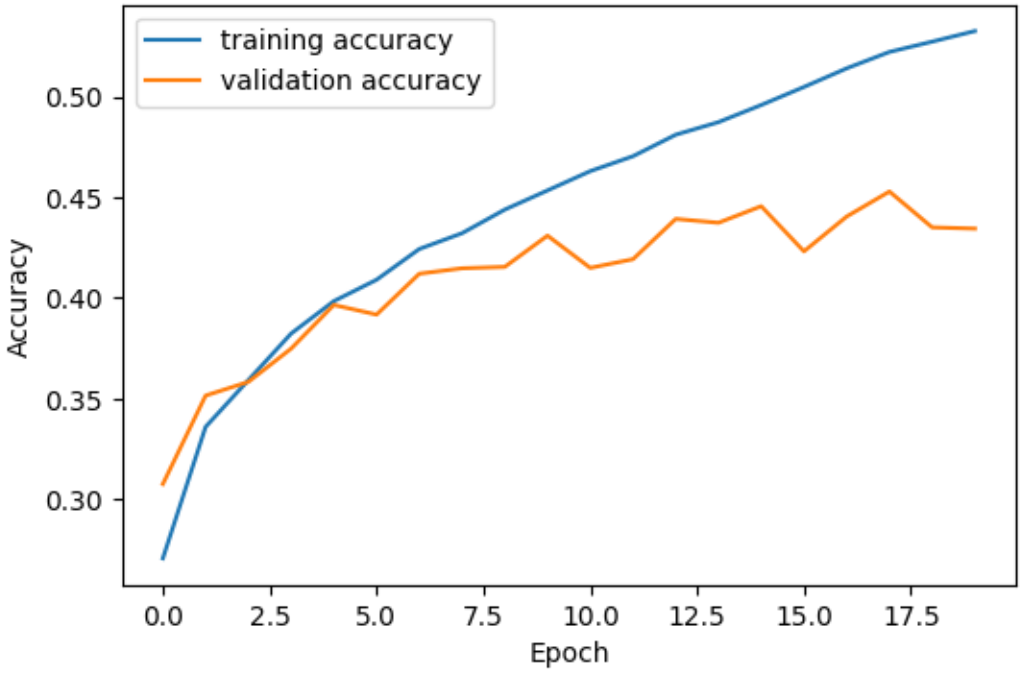
می بینیم که اضافه کردن لایه مخفی تاثیری در افزایش دقت شبکه ندارد، حال از شبکه اولیه یک لایه مخفی کم می کنیم

جدول ۱-۹ معماری شبکه

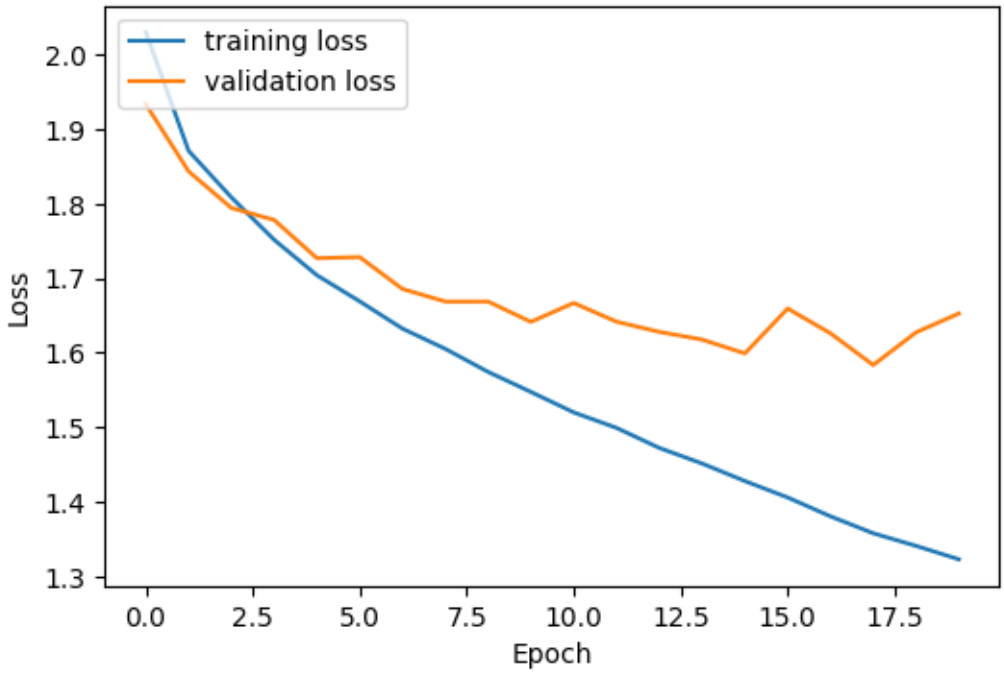
```
Model: "sequential_39"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_164 (Dense)           (None, 1024)              1049600

 dense_165 (Dense)           (None, 512)               524800

 dense_166 (Dense)           (None, 10)                5130

=================================================================
Total params: 1,579,530
Trainable params: 1,579,530
Non-trainable params: 0
_________________________________________________________________
```

```
training time: 41.528897523880005
```

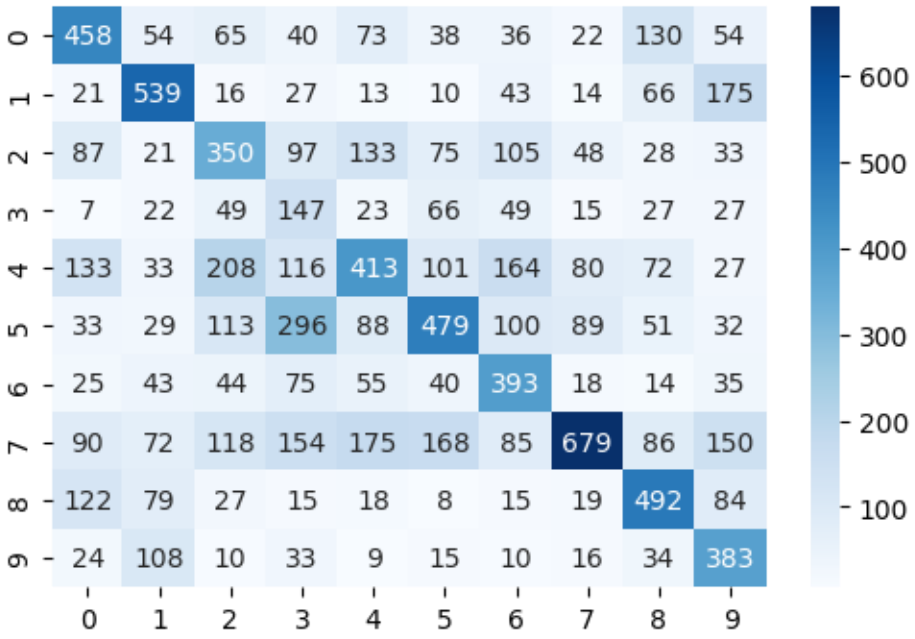
شکل ۱-۹۸ زمان آموزش



شکل ۱-۹۹ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۰۰ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۱-۱۰۱ ماتریس آشفتگی

```
accuracy on test data: 0.4332999885082245
loss on test data: 1.6419756412506104
f1_score on test data: 0.4281711536691701
recall on test data: 0.4333
presicion on test data: 0.4494326136314411
```

شکل ۱-۱۰۲ معیار های مختلف برای داده های تست

با توجه به نتایج بالا می بینیم که با افزایش تعداد لایه مخفی زمان آموزش افزایش میابد، دقت مدل نیز برای ۳ لایه مخفی بیشترین مقدار را دارد و پس از آن تفاوت چشمگیری در دقت مدل حاصل نمی شود. زیرا برای پیچیدگی یادگیری طبقه بندی این دیتاست ۳ لایه مخفی کافی می باشد.

Hidden Layer (HL)

3 HL > 4 HL > 5 HL > 2 HL

ط)

با توجه به نتایج پیشین بهترین پارامتر ها عبارتند از

تعداد نورون: دو لایه برابر ۱۰۲۴ به دلیل اینکه ورودی های شبکه بردار های ۱۰۲۴ تایی هستند، با توجه به نتایج تعداد نورون کمتر باعث کاهش دقت مدل و افزایش نورون تاثیر چشمگیری در افزایش دقت ندارد اما باعث افزایش زمان آموزش می شود

سایز بسته: برابر ۱۲۸ چون هم نسبت به سایر سایز های بسته دقت بیشتری به ما می دهد و هم زمان آموزش کمتری نسبت به بسته های کوچکتر نیاز دارد. سایز کمتر باعث نویزی شدن یادگیری و کند شدن آموزش می شود، و سایز بسته بزرگتر امکان تعمیم مدل را کاهش می دهد.

تابع فعالساز: ReLU برای لایه های مخفی، چون زمان آموزش کمتری نیاز دارد (به دلیل کم هزینه بودن محاسبه مشتق آن) نسبت به توابع دیگر و نداشتن مشکلاتی از جمله ناپدید شدن گرادیان، همچنین دقت بهتری را از بقیه حاصل می سازد

تابع خطا: Cross Entropy چون نسبت به دو تابع دیگر زمان آموزش کمتر و دقت بیشتری را حاصل می سازد و معمولا برای طبقه بندی های چند کلاسه کاربرد دارد

روش بهینه سازی: SGD همراه با تکانه چون زمان آموزش خیلی کمتری را به همراه دقت بیشتر نسبت به دو روش دیگر حاصل می سازد. این روش با توجه به تکانه ای که دارد سرعت همگرایی بیشتری دارد

تعداد لایه مخفی: ۳ لایه مخفی، چون دقت بیشتری را نسبت به ۲ لایه مخفی حاصل می سازد و همچنین نسبت به لایه های مخفی بیشتر زمان آموزش کمتری نیاز دارد. پیچیدگی مسئله در حدی است که سه لایه مخفی برای آموزش مدل کافی می باشد.

ن)

جدول ۱-۱۰ معماری و جزییات مدل

تعداد لایه مخفی	۳
تعداد نورون هر لایه مخفی	۲ لایه با ۱۰۲۴ نورون ۱ لایه با ۵۱۲ نورون
تابع فعالساز لایه های مخفی	ReLU
تابع فعالساز لایه خروجی	Softmax
تابع خطا	Cross Entropy
سایز بسته	۱۲۸
روش بهینه سازی	SGD with Momentum

جدول ۱-۱۱ عملکرد مدل

مدت زمان آموزش	۵۳ ثانیه
خطای داده آموزش	۱.۱۹
دقت داده آموزش	۵۹٪
دقت پیش بینی داده تست	۴۵.۳٪
خطای داده تست	۱.۶۱
Precision	۴۶.۱٪
Recall	۴۵.۳٪
F1 score	۴۵٪

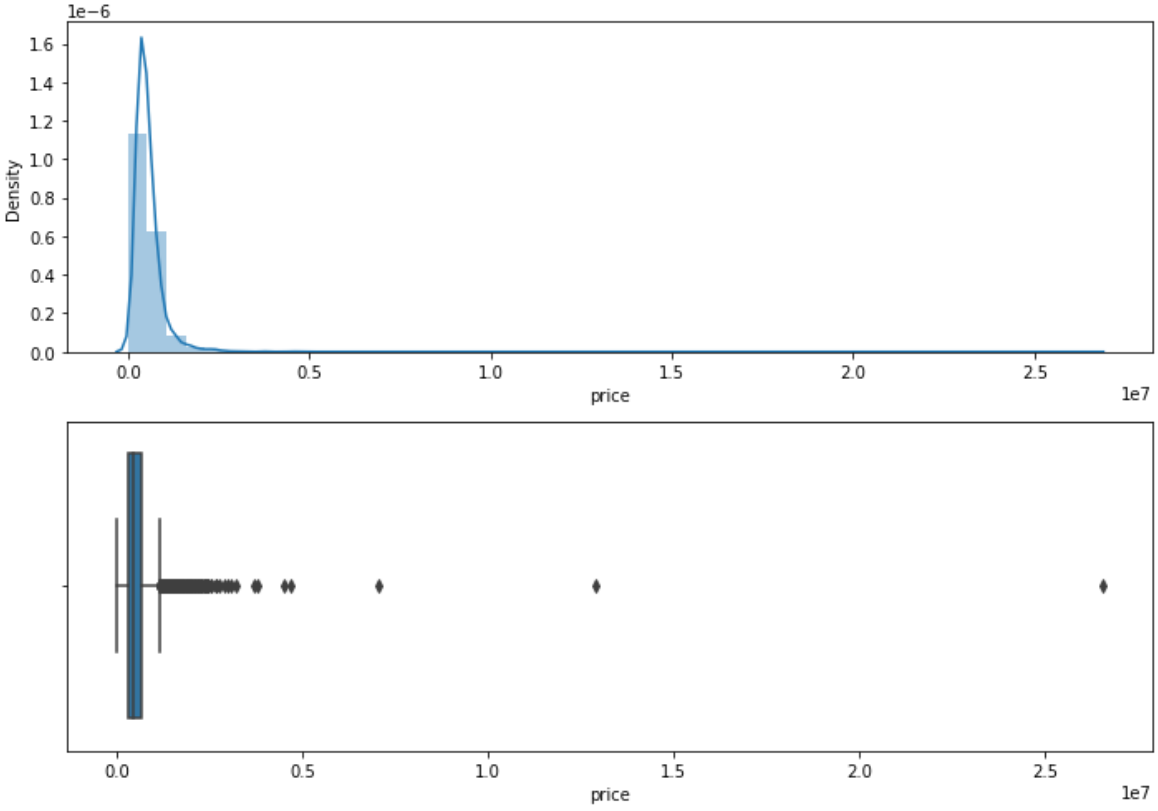
مقادیر بالا میانگین ۱۰۰ بار اجرای الگوریتم است.

مشخصات سیستم استفاده شده: RAM: 12GB, GPU: Tesla K80

سوال دوم – MLP(Regression)

الف)

- به عنوان پیش پردازش، داده ها را استاندارد می کنیم، یعنی با اعمال تبدیل مناسب داده های عددی را به میانگین صفر و واریانس ۱ می بریم.
- سپس ستون مربوط به تاریخ را به ۳ ستون عددی سال، ماه و روز تبدیل می کنیم تا برای مدل ما قابل استفاده باشد
- سپس داده های غیر عددی را به صورت one-hot کد می کنیم تا برای مدل ما قابل استفاده باشد
- و در نهایت داده های پرت (با قیمت خیلی زیاد) را از دادگان حذف می کنیم تا باعث ایجاد خطا در آموزش نشود. توزیع قیمت خانه ها به صورت زیر می باشد.



شکل ۲-۱ توزیع قیمت خانه ها

با توجه به توزیع بالا خانه های با قیمت بیشتر از 10000000 را حذف می کنیم (۲ داده).

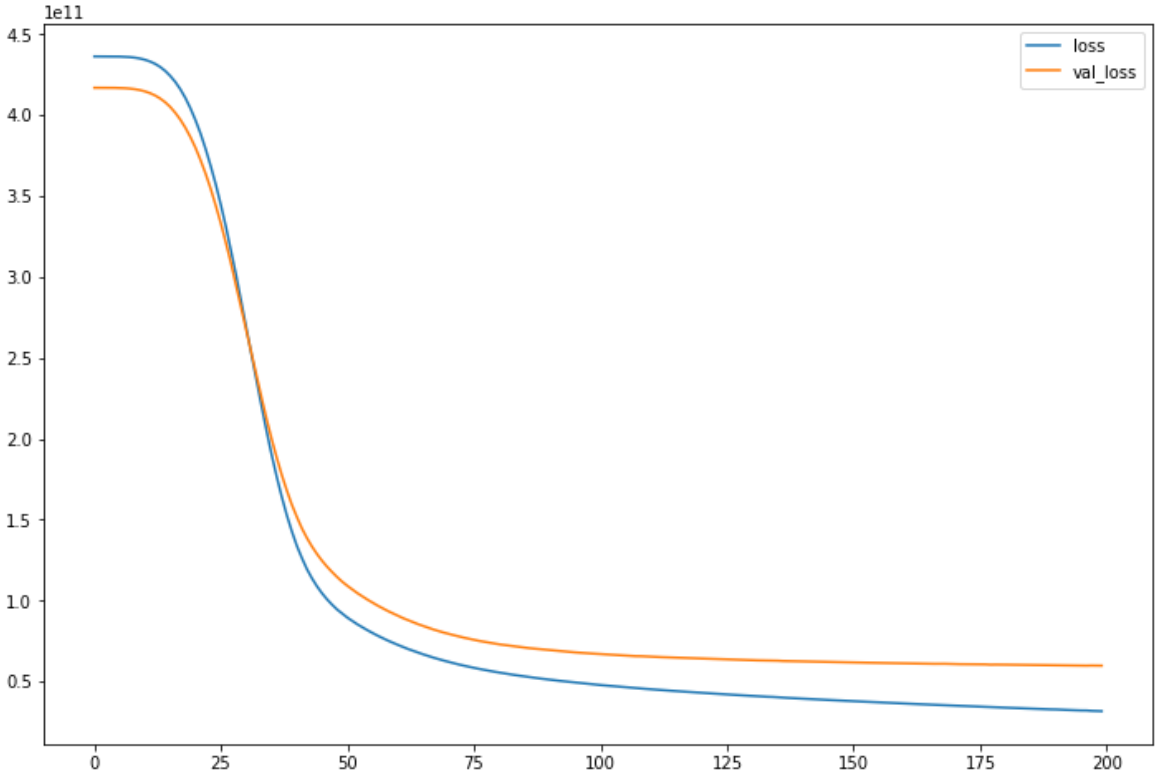
ب)

حال به طراحی چند مدل می پردازیم تا تعداد لایه مناسب و تعداد نورون و تابع فعالساز را تعیین کنیم. همچنین از روش بهینه سازی Adam همراه با سایز بسته ۱۲۸ استفاده می کنیم.

معماری مدل اول با ۳ لایه مخفی تماما متصل و تابع فعالساز ReLU و تعداد نورون ۲۰ در هر لایه

جدول ۲-۱ معماری شبکه

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_50 (Dense)             (None, 20)                93220

 dense_51 (Dense)             (None, 20)                420

 dense_52 (Dense)             (None, 20)                420

 dense_53 (Dense)             (None, 1)                 21

=================================================================
Total params: 94,081
Trainable params: 94,081
Non-trainable params: 0
_________________________________________________________________
```

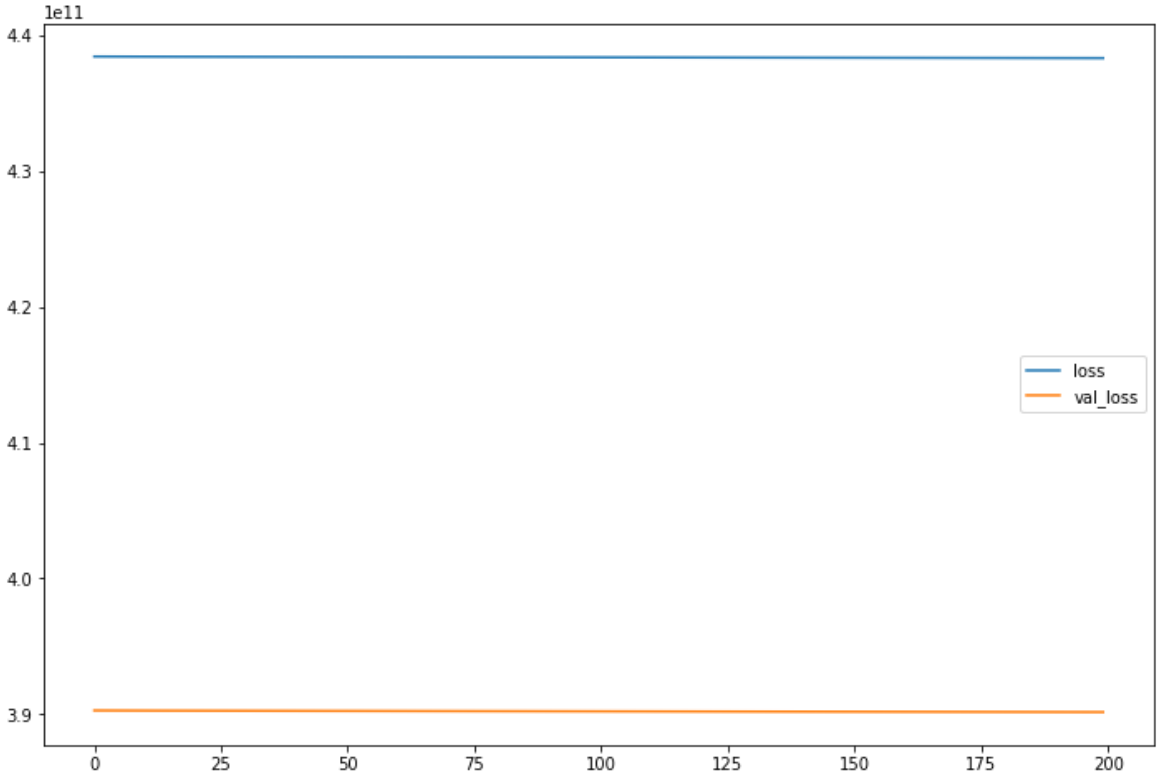


شکل ۲-۲ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

```
MAE: 109045.90536710055
MSE: 32629780359.696095
RMSE: 180637.15110601168
VarScore: 0.7513030243599734
```

شکل ۲-۳ معیار های مختلف برای داده های تست

حال مدل دوم با ۳ لایه مخفی تماما متصل و تابع فعالساز Tanh و تعداد نورون ۲۰ در هر لایه



شکل ۲-۴ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

```
MAE: 549635.6106047031
MSE: 439121256767.1287
RMSE: 662662.2493903879
VarScore: 5.046962847643499e-11
```

شکل ۲-۵ معیار های مختلف برای داده های تست

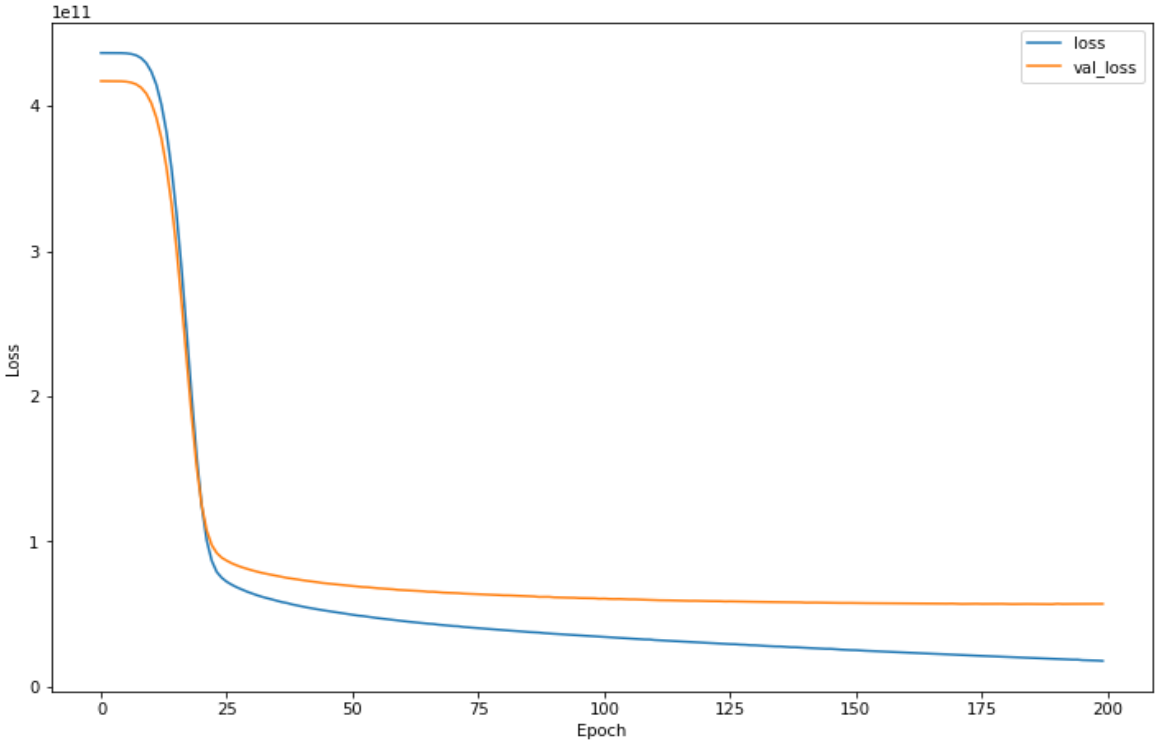
با توجه به نتایج بالا متوجه می شویم که تابع فعالساز Tanh برای آموزش مناسب نیست زیرا به دلیل مشکل کوچک شدن گرادیان نمی تواند شبکه را آموزش دهد. پس در ادامه از ReLU استفاده می کنیم

حال مدل سوم با با ۴ لایه مخفی تماما متصل و تابع فعالساز ReLU و تعداد نورون ۲۰ در هر لایه

جدول ۲-۲ معماری شبکه

Layer (type)	Output Shape	Param #
dense_45 (Dense)	(None, 20)	93220
dense_46 (Dense)	(None, 20)	420
dense_47 (Dense)	(None, 20)	420
dense_48 (Dense)	(None, 20)	420
dense_49 (Dense)	(None, 1)	21

Total params: 94,501
Trainable params: 94,501
Non-trainable params: 0



شکل ۲-۶ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

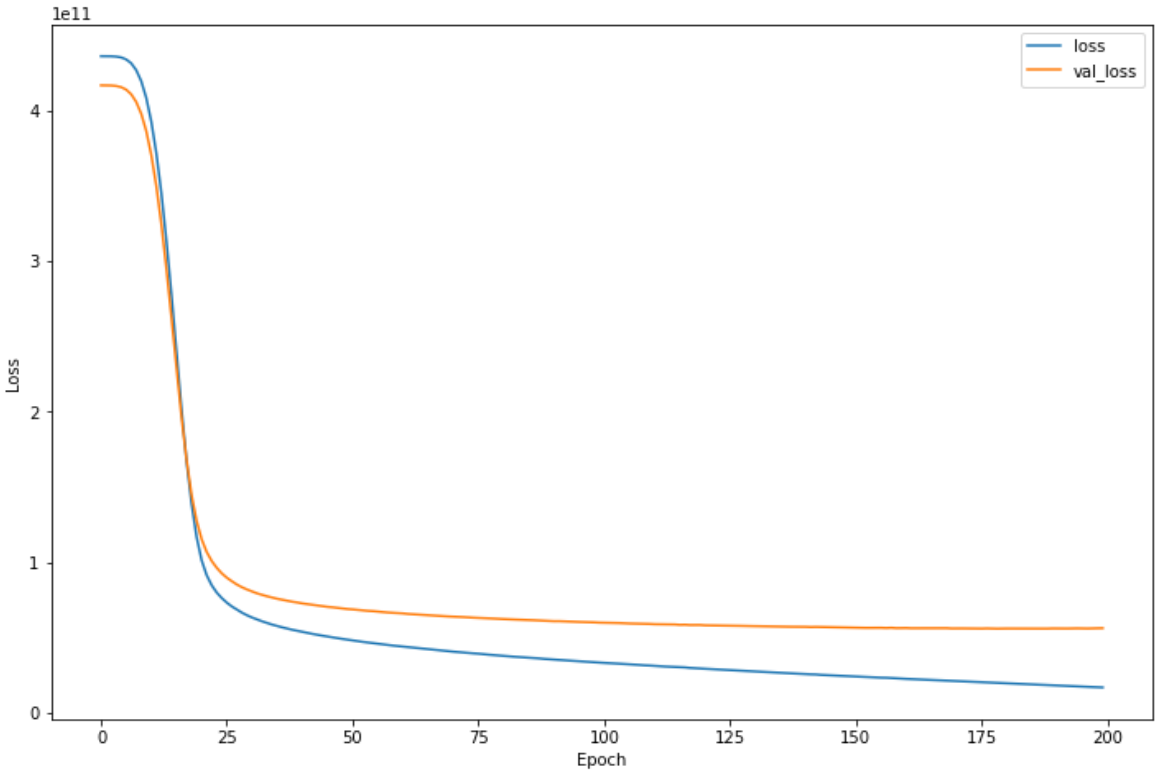
```
MAE: 112055.07714673423
MSE: 30653457516.39749
RMSE: 175081.28831030885
VarScore: 0.7690804958817874
```

شکل ۲-۷ معیار های مختلف برای داده های تست

حال مدل چهارم با با ۳ لایه مخفی تماما متصل و تابع فعالساز ReLU و تعداد نورون ۶۰ در هر لایه

جدول ۲-۳ معماری شبکه

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_54 (Dense)             (None, 60)                279660

 dense_55 (Dense)             (None, 60)                3660

 dense_56 (Dense)             (None, 60)                3660

 dense_57 (Dense)             (None, 1)                 61

=================================================================
Total params: 287,041
Trainable params: 287,041
Non-trainable params: 0
_________________________________________________________________
```



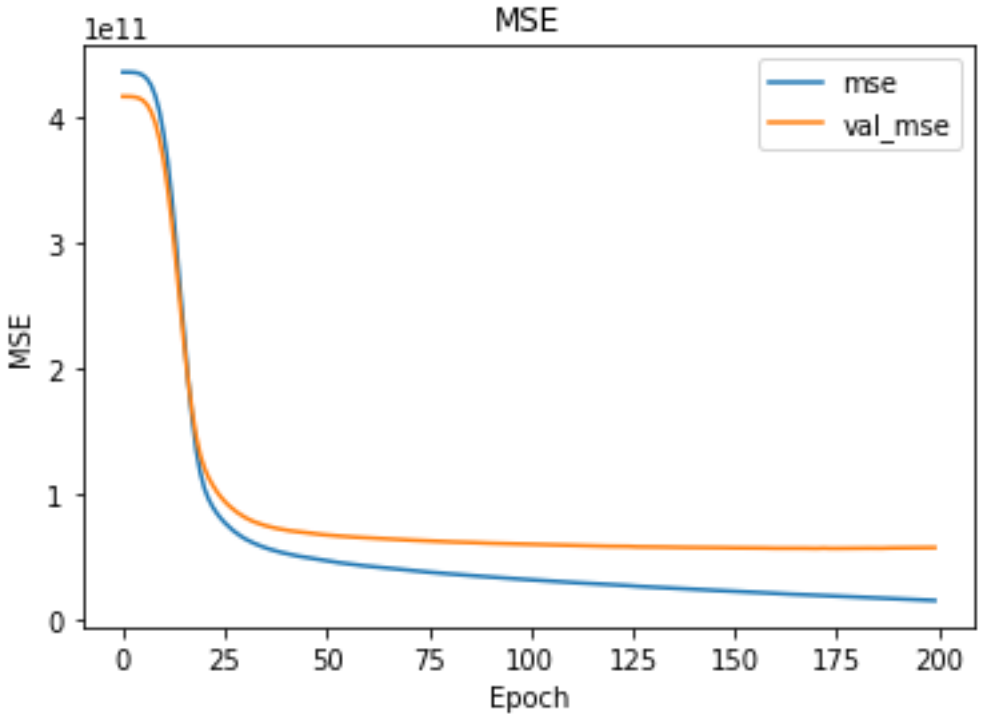
شکل ۲-۸ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

```
MAE: 109855.32374069565
MSE: 30201849626.516754
RMSE: 173786.793590643
VarScore: 0.77352949749968
```

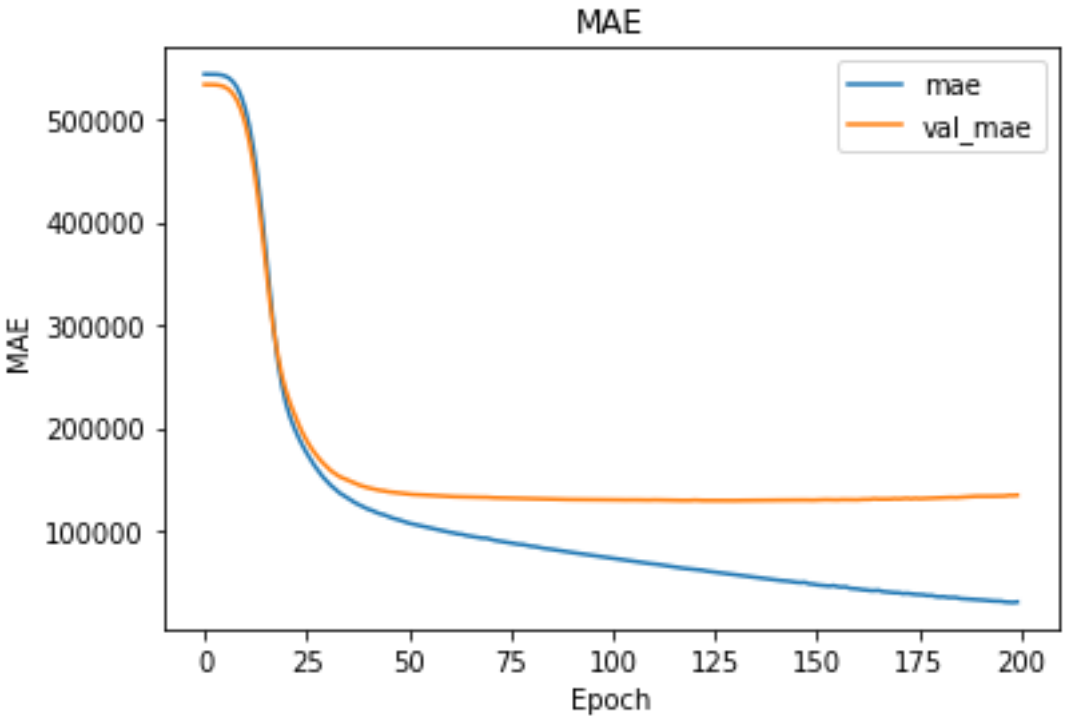
شکل ۲-۹ معیار های مختلف برای داده های تست

با توجه به نتایج بالا متوجه می شویم که افزایش تعداد لایه ها به ۴ و افزایش تعداد نورون های هر لایه از ۲۰ به ۶۰ اندکی باعث بهبود عملکرد مدل می شود، همچنین تابع فعالساز ReLU نسبت به سایر توابع فعالساز عملکرد بهتری دارد. پس در ادامه با ۳ لایه مخفی ۶۰ نورونه با تابع فعالساز ReLU کار خواهیم کرد.

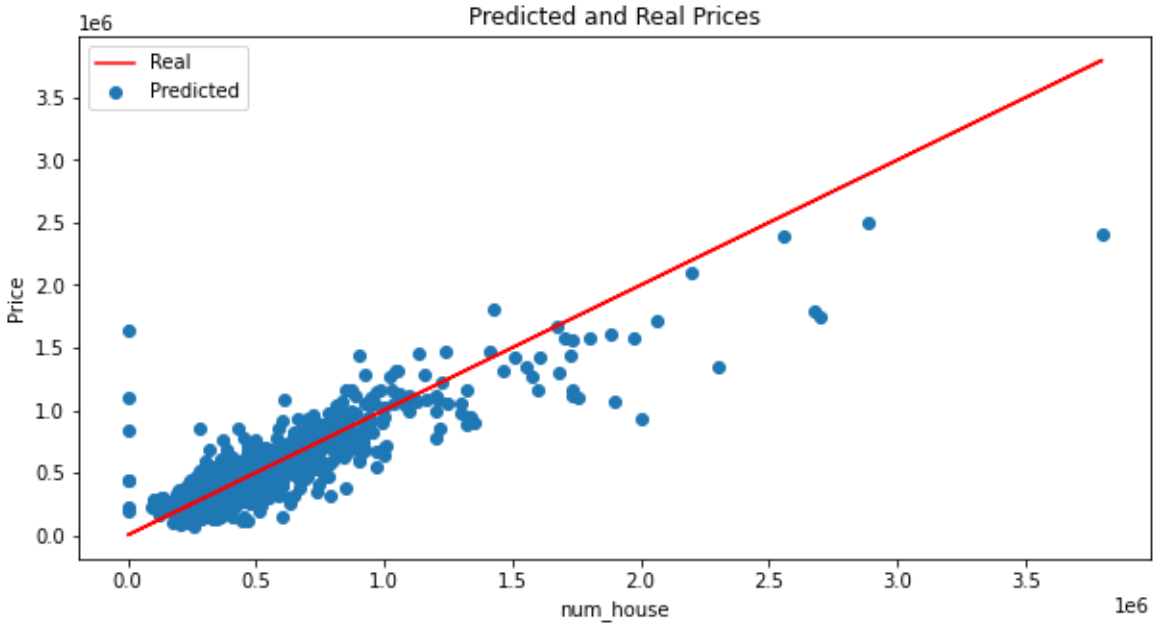
ج)



شکل ۲-۱۰ MSE بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۲-۱۱ MAE بر حسب ایپاک برای داده های آموزش و ارزیابی

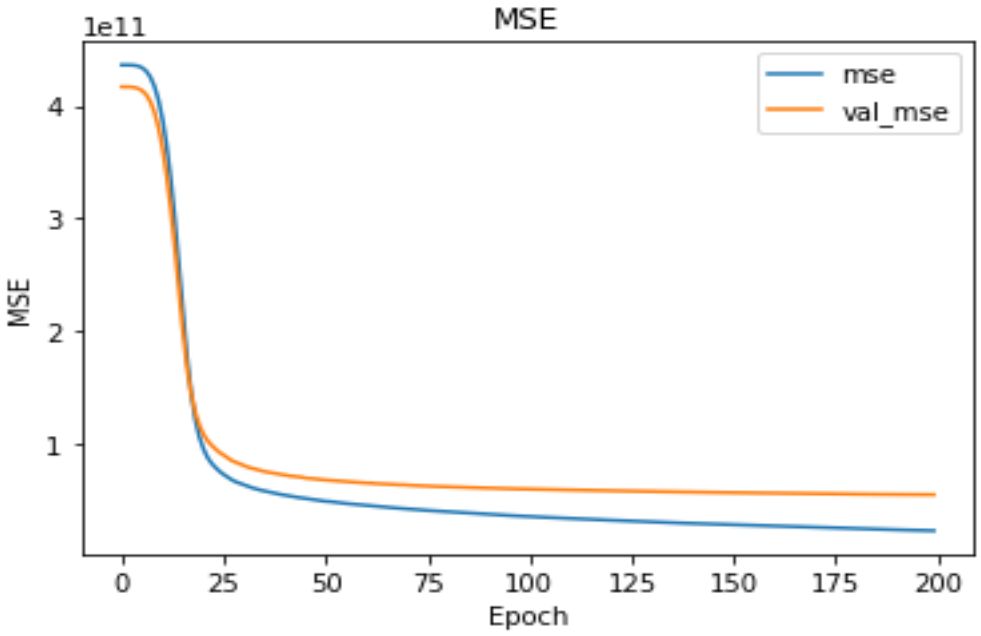


شکل ۲-۱۲ داده های پیش بینی شده به رنگ آبی در مقابل قیمت واقعی روی خط قرمز رنگ

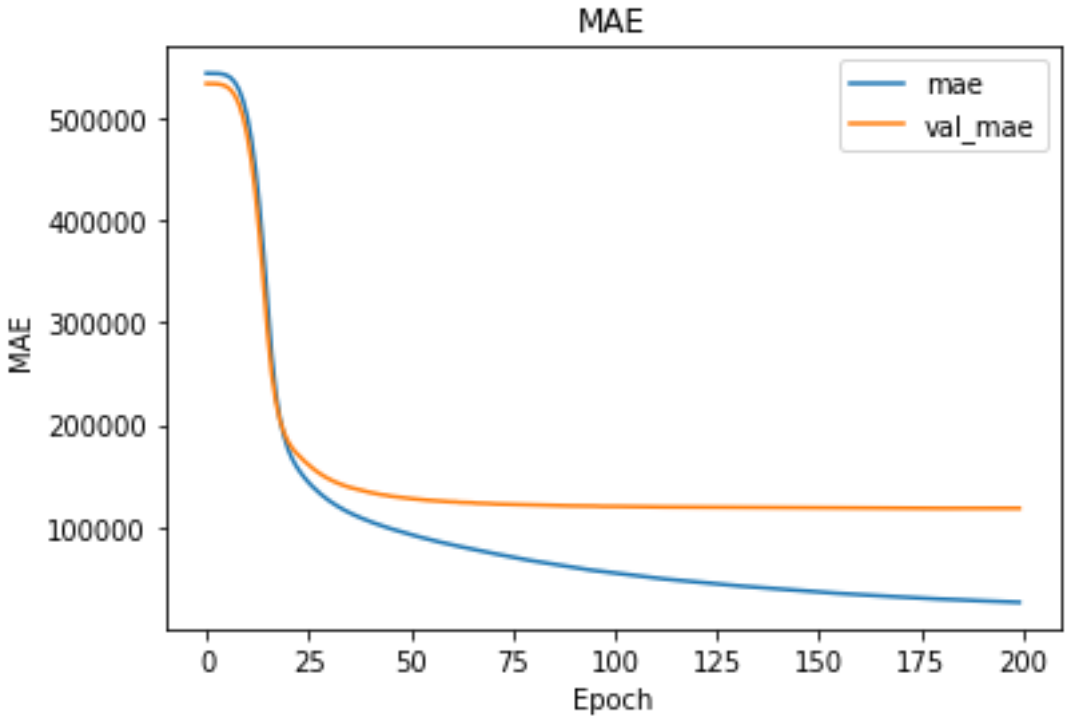
```
MAE: 116901.30270148996
MSE: 32448035841.150127
RMSE: 180133.38347222074
VarScore: 0.7584857883429089
```

شکل ۲-۱۳ معیار های مختلف برای داده های تست

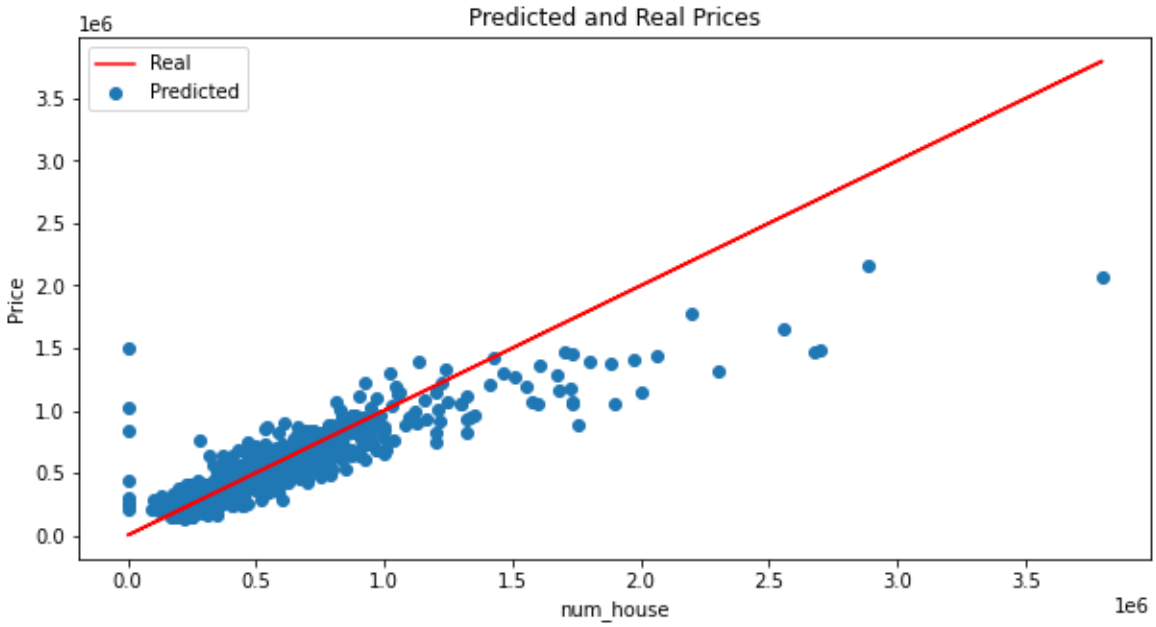
د)



شکل ۲-۱۴ MSE بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۲-۱۵ MAE بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۲-۱۶ داده های پیش بینی شده به رنگ آبی در مقابل قیمت واقعی روی خط قرمز رنگ

```
MAE: 97560.6018288299
MSE: 31378287907.589947
RMSE: 177139.17665945596
VarScore: 0.7658932304120495
```

شکل ۲-۱۷ معیار های مختلف برای داده های تست

۵)

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

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

با توجه به روابط ریاضی می بینیم که معیار MAE در واقع برابر میانگین اندازه خطای پیش بینی است و همچنین MSE برابر میانگین مربع خطای پیش بینی است.

نمودارهای MSE و MAE بر حسب ایپاک برای دو حالتی که loss را MSE و MAE در نظر گرفتیم، تفاوت چندانی ندارد، اما خطای MSE و MAE و امتیاز واریانس وقتی از تابع خطای MAE استفاده می کنیم بهتر هستند. دلیل این است که در معیار MSE مربع خطا در نظر گرفته می شود و برای داده های پرت این خطا مقدار بزرگی می شود و وزن های مدل بابت آن را تغییر زیادی می کند که باعث می شود

قدرت تعمیم مدل نسبت به معیار MAE کمتر باشد. با توجه به نمودار ها حدود ۸۰ ایپاک بهترین مقدار است چون پس از آن مقدار خطای داده های ارزیابی کاهش نمی یابد.

امتیازی

رگرسیون خطی یک مدل آماری برای پیش‌بینی یک یا چند متغیر از روی یک یا چند متغیر دیگر است. به متغیرهایی که پیش‌بینی بر روی آن انجام می‌شود متغیر وابسته و به متغیرهایی که پیش‌بینی به کمک آن‌ها انجام می‌شود متغیرهای مستقل می‌گویند. در بسیاری از مسائل رایج رگرسیون، ورودی چندمتغیره است. اگر فرض کنیم متغیر ما p بعد دارد یعنی $\vec{x} = [x_1, x_2, \dots, x_p]$ مسئله رگرسیون به یک مسئله بهینه‌سازی برای پیدا کردن $p + 1$ پارامتر تبدیل می‌شود، به این معنی که ما یک پارامتر چندمتغیره به اسم $\vec{\beta} = [\beta_0, \beta_1, \dots, \beta_p]$ داریم و سعی می‌کنیم که متغیر وابسته که همان y است را با بردار $\vec{x}$ تخمین بزنیم. در چنین مدلی پارامتر بهینه آن پارامتری است که یک تابع هزینه(خطای حداقل مربعات) را به حداقل برساند و تخمین‌های ما را به متغیر وابسته بسیار نزدیک کند. حساب پارامتر بهینه عبارت است از:

$$\vec{\beta} = \underset{\vec{\beta}}{argmin} \sum_{i=1}^{n} (\vec{\beta}.\overrightarrow{x_i} - y_i)^2$$

برای حل این معادله می توان از روش گرادیان نزولی تصادفی[1] یا تحلیلی به صورت زیر استفاده کرد

$$\vec{\beta} = (X^T X)^{-1} X^T Y$$

تابع هزینه برای مدل Ridge به صورت زیر می باشد که تفاوت آن با تابع هزینه مدل رگرسیون خطی اضافه شدن یک ترم رگولاتور $\lambda \sum_{j=1}^{p} \beta_j^2$ می باشد که بزرگ شدن وزن ها را جریمه می کند.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

بنابر این مدل Ridge از بزرگ شدن بیش از حد وزن ها و در نتیجه از بیش برازش جلوگیری می کند.

[1] Stochastic Gradient Descent

```
Ridge MAE: 103140.33366868865
Ridge MSE: 29720926931.508945
Ridge RMSE: 172397.58389115825
Ridge VarScore: 0.7710464984575975

Linear Regression MAE: 794691607220.3866
Linear Regression MSE: 9.502264494466243e+24
Linear Regression RMSE: 3082574329106.4766
Linear Regression VarScore: -73197704121424.52
```

شکل ۲-۱۸ معیار های مختلف برای روش Ridge در مقایسه با Linear Regression

با مشاهده نتایج بالا می بینیم که مدل Ridge عملکرد خیلی بهتری نسبت به مدل رگرسیون خطی معمولی دارد، زیرا در مواردی که برچسب ها زیاد است و کاهش بعد انجام نشده است از بیش برازش جلوگیری می کند و مدل را به خوبی آموزش می دهد، عملکرد مدل Ridge تقریبا با عملکرد شبکه عمیقی که طراحی کردیم برابر است.

سوال ۳ ـ کاهش ابعاد

الف)

ایده پشت تجزیه مؤلفه اصلی[1] (PCA) پیدا کردن جهت هایی با بیشترین واریانس داده در آن جهت است تا با تصویر کردن داده ها روی این جهت ها باعث کاهش ابعاد داده شود. در ادامه روش انجام این کار را توضیح می دهیم.

فرض می کنیم بردار $X_{n\times m}$ را داریم

۱. استاندارد کردن داده ها (تبدیل به میانگین صفر و وارینس ۱)

$$B = \frac{X - \mu_x}{\sigma}$$

که μ_x میانگین x و σ انحراف معیار است

۲. به دست آوردن ماتریس کواریانس

$$C = \frac{1}{n-1} X^T . X$$

۳. تجزیه مقادیر ویژه ماتریس کواریانس

$$C = VDV^{-1}$$

۴. انتخاب k بردار ویژه متناظر با k بزرگترین مقدار ویژه

[1] Principal Component Analysis

مقادیر ویژه روی قطر های ماتریس D را مرتب کرده و k مقدار بردار ویژه متناظر با k بزرگترین مقدار ویژه را انتخاب می کنیم، این بردار ویژه ها در واقع همان جهت هایی هستند که داده ها در آن جهت بیشترین واریانس را دارند.

۵. درست کردن ماتریس W از بردار های ویژه پیدا شده

$$W = [w_1, w_2, \dots, w_k]$$

۶. تصویر کردن داده ها روی فضای k-بعدی با ضرب کردن W در آنها

$$X'_{n\times k} = X_{n\times m} W_{m\times k}$$

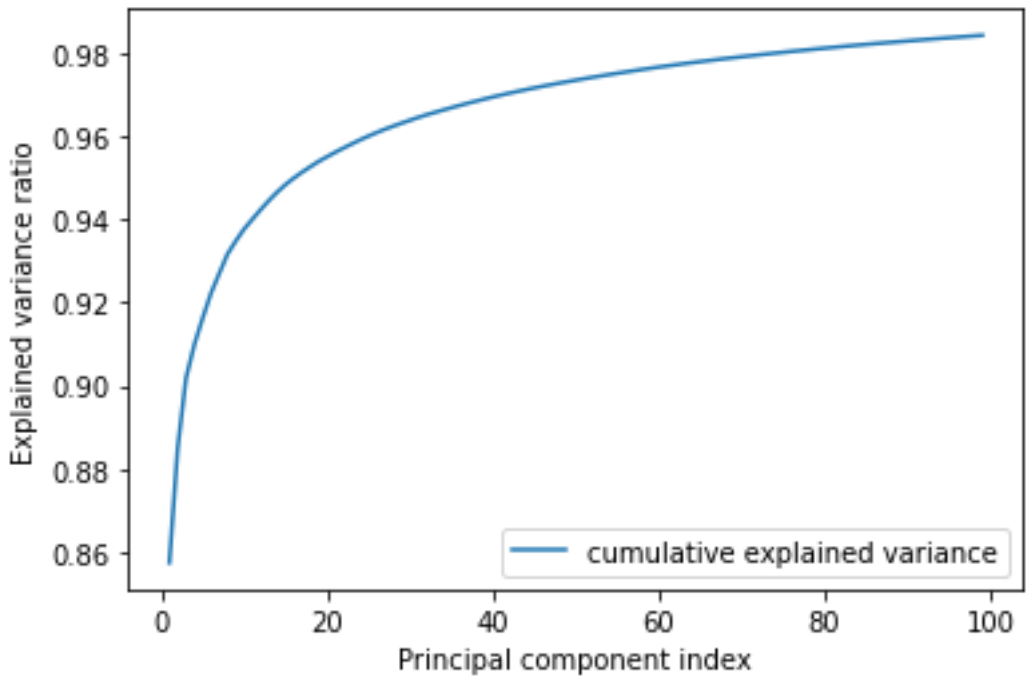
بردار X با بعد $n \times m$ به بردار X' با بعد $n \times k$ کاهش یافت.

روش دوم:

می دانیم تجزیه مقادیر تکین ماتریس A در واقع برابر تجزیه مقادیر ویژه ماتریس $A^T A$ می باشد. پس یعنی تجزیه SVD داده های $X_{n\times m}$ در واقع همان تجزیه مقادیر ویژه ماتریس کواریانس آن هستند. پس کاهش بعد به صورت زیر انجام می شود.

$$\text{SVD decomposition: } X = U\Sigma V^T$$

$$X'_{n\times k} = \Sigma V$$



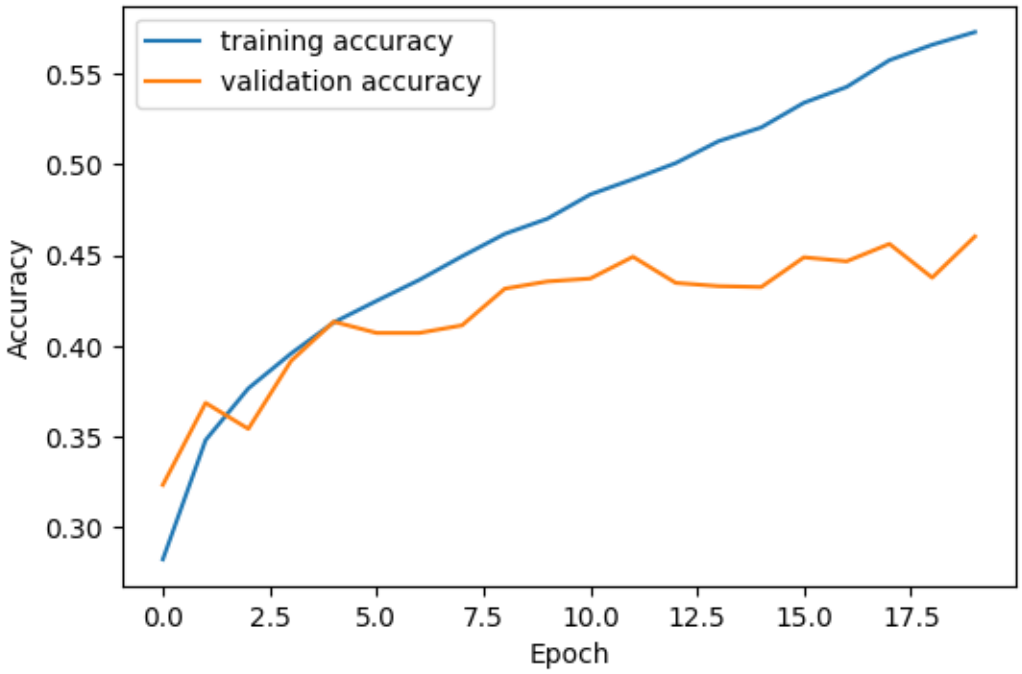
شکل ۳-۱ توزیع واریانس بر حسب تعداد مولفه های اصلی

با توجه به نمودار شکل؟ می بینیم که با حدود ۸۰ مؤلفه اصلی حدود ۹۸٪ واریانس داده ها را پوشش می دهیم، و مقدار زیادی از اطلاعات مفید تصاویر را می توانیم در بعد ۸۰ ذخیره کنیم.

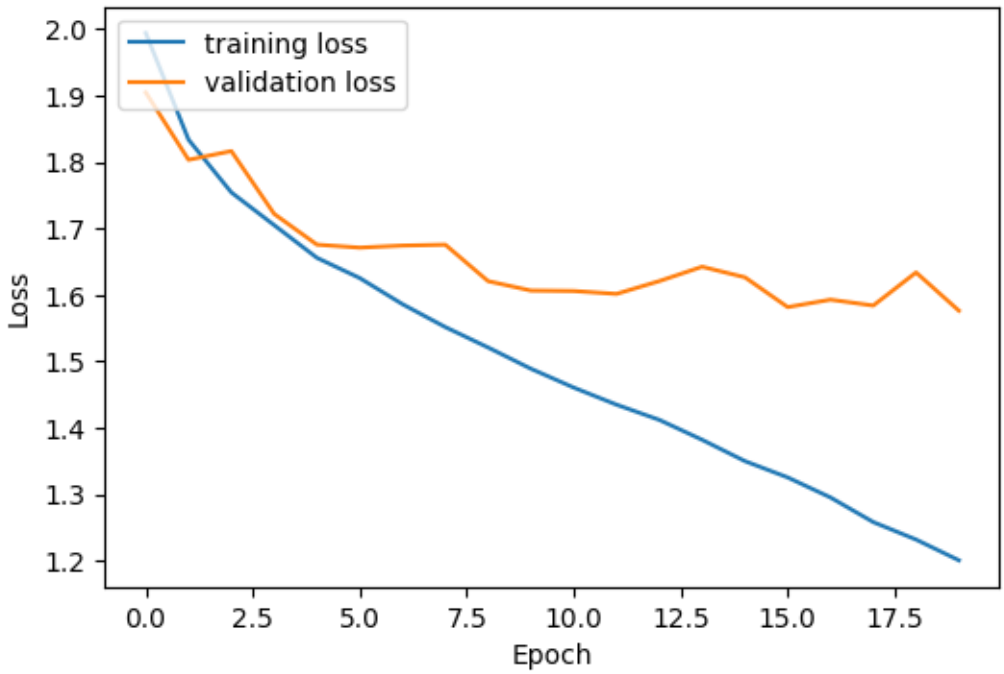
این تبدیل را اعمال کرده و بهترین مدل به دست آمده از قسمت قبل را با تصاویر کاهش بعد یافته در ۲۰ ایپاک آموزش می دهیم.

جدول ۳-۱ معماری شبکه

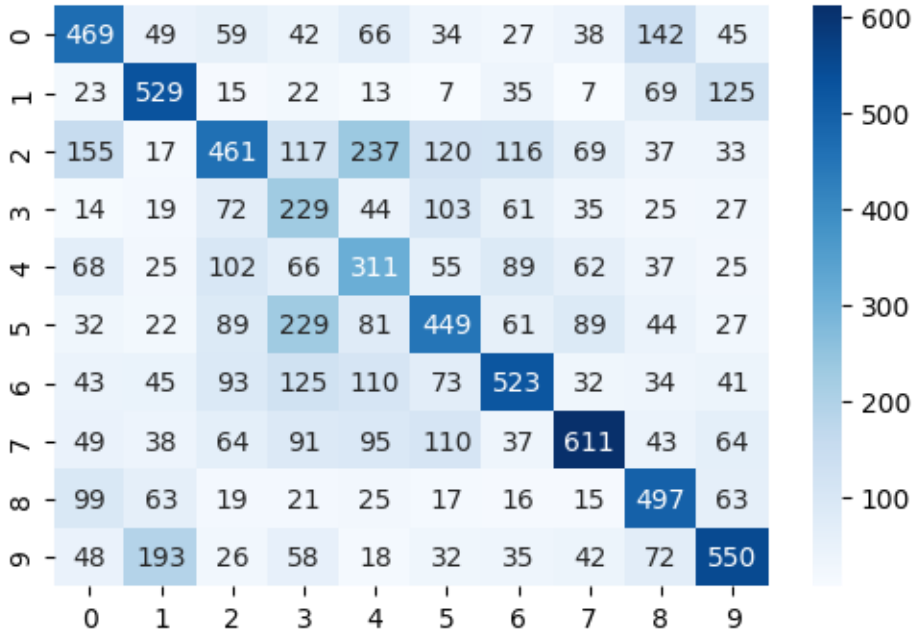
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense (Dense)                (None, 1024)              79872

 dense_1 (Dense)              (None, 1024)              1049600

 dense_2 (Dense)              (None, 512)               524800

 dense_3 (Dense)              (None, 10)                5130

=================================================================
Total params: 1,659,402
Trainable params: 1,659,402
Non-trainable params: 0
_________________________________________________________________
```



شکل ۳-۲ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۳-۳ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۳-۴ ماتریس آشفتگی

```
accuracy on test data: 0.4629000127315521
loss on test data: 1.5613257884979248
f1_score on test data: 0.4602517930421846
recall on test data: 0.4629
presicion on test data: 0.46646604213946546
```

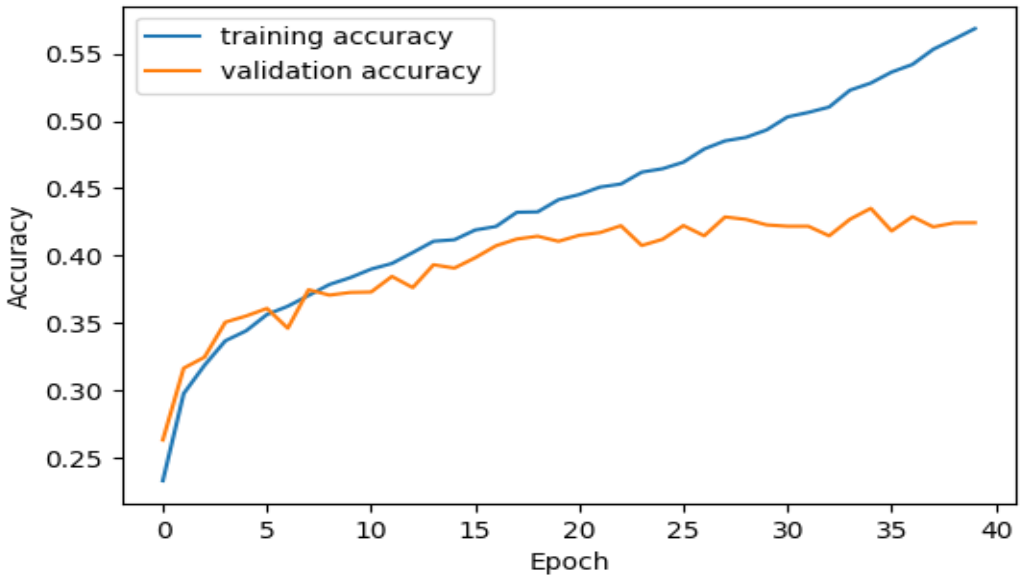
شکل ۳-۵ معیار های مختلف برای داده های تست

می بینیم که استفاده از این روش کاهش بعد، علاوه بر کاهش زمان آموزش باعث افزایش دقت و سایر معیار های شبکه نیز شده است.

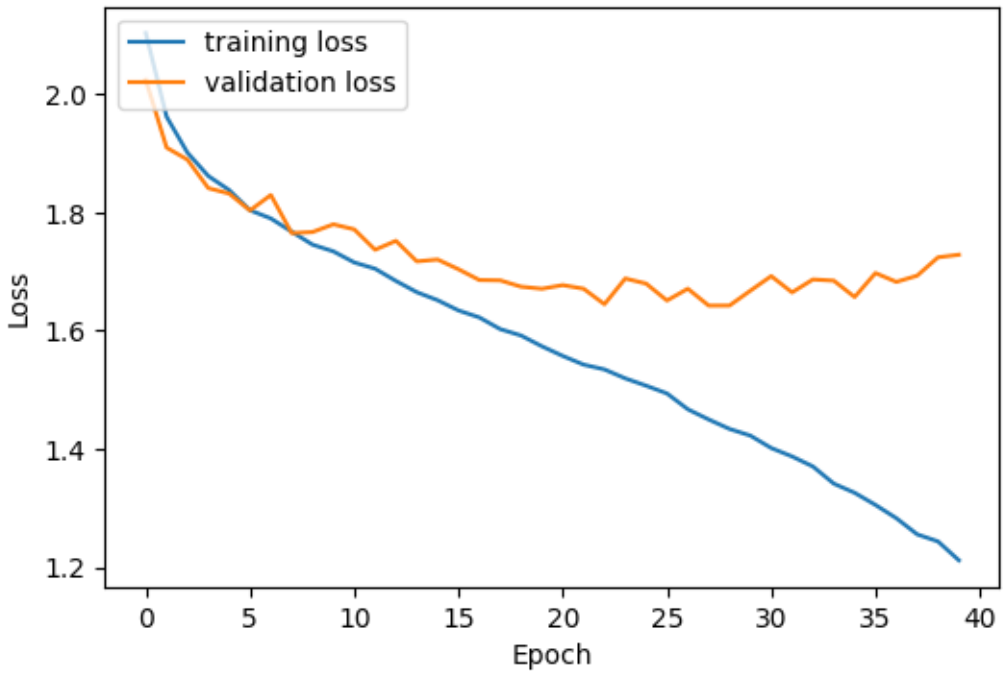
ب)

جدول ۳-۲ معماری انکودر

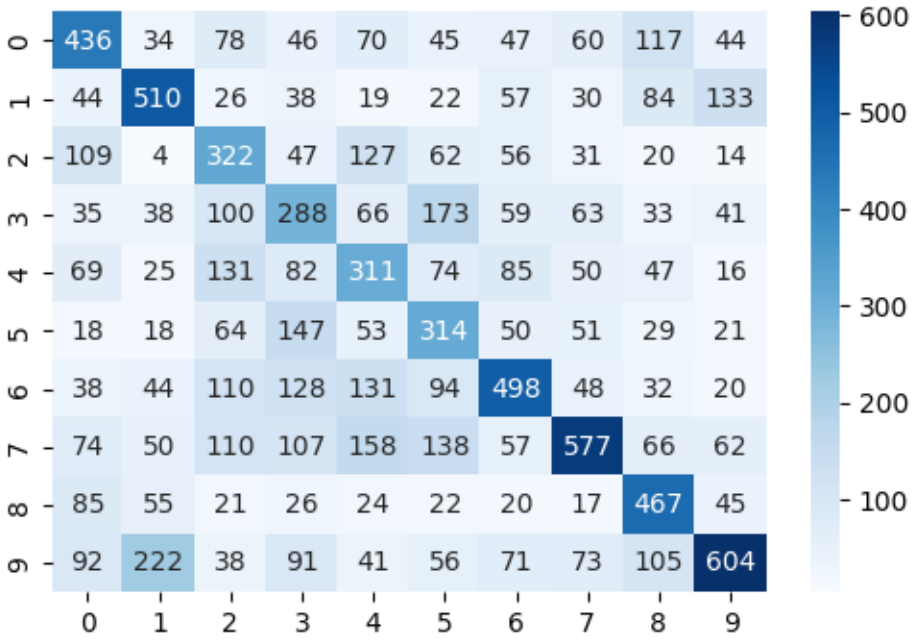
```
 Layer (type)                 Output Shape              Param #
=================================================================
 input_8 (InputLayer)         [(None, 1024)]            0

 dense_63 (Dense)             (None, 1024)              1049600

 dense_64 (Dense)             (None, 77)                78925

 dense_65 (Dense)             (None, 512)               39936

 dense_66 (Dense)             (None, 1024)              525312

=================================================================
Total params: 1,693,773
Trainable params: 1,693,773
Non-trainable params: 0
_________________________________________________________________
```



شکل ۳-۶ دقت بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۳-۷ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی



شکل ۳-۸ ماتریس آشفتگی

جدول ۳-۳ مقایسه مدل سوال ۱ با مدل کاهش بعد یافته به کمک دو روش PCA و Autoencoder

	مدل سوال ۱	با PCA	با Autoencoders
مدت زمان آموزش	۵۳ ثانیه	۴۳ ثانیه	۸۳ ثانیه
خطای داده آموزش	۱.۱۹	۱.۲۰	۱.۲۱
دقت داده آموزش	۵۹٪	۵۸٪	۵۷٪
دقت پیش بینی داده تست	۴۵.۳٪	۴۶.۲٪	۴۳.۲٪
خطای داده تست	۱.۶۱	۱.۵۶	۱.۶۹٪
Precision	۴۶.۱٪	۴۶.۶٪	۴۳.۴٪
Recall	۴۵.۳٪	۴۶.۳٪	۴۳.۲٪
F1 score	۴۵٪	۴۶٪	۴۲.۸٪

```
accuracy on test data: 0.43270000815391154
loss on test data: 1.6933985948562622
f1_score on test data: 0.42835041857105516
recall on test data: 0.4327
presicion on test data: 0.4342665284202405
```

شکل ۳-۹ معیار های مختلف برای داده های تست

ج)

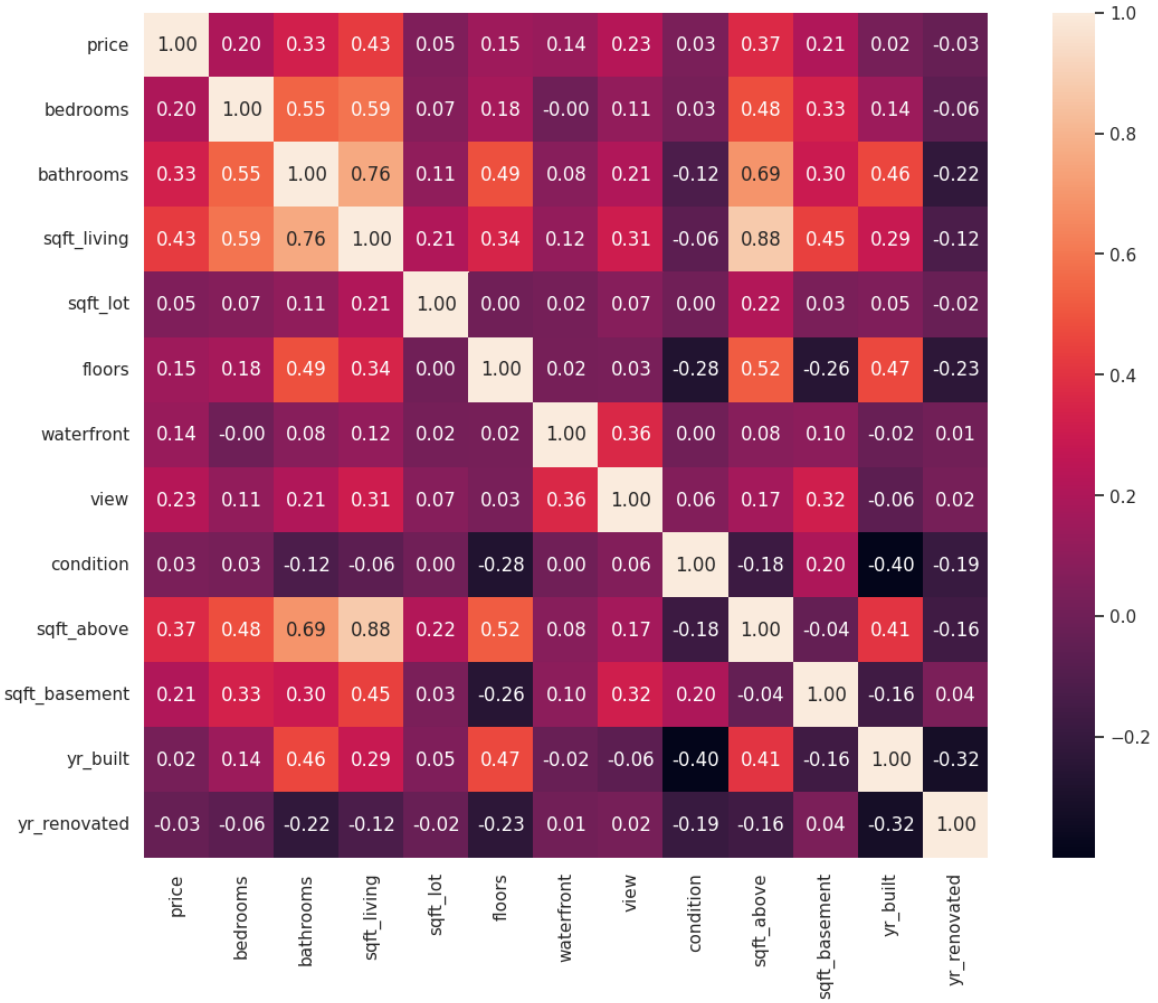
با توجه به نتایج به دست آمده می بینیم که دقت و معیار های مدل با استفاده از PCA نسبت به قبل افزایش داشته است(به دلیل کاهش بعد و جلوگیری از بیش برازش) اما در حالت استفاده از Autoencoder ها دقت کاهش یافته است، دلیل آن می تواند عدم استخراج ویژگی ها به خوبی روش سیستماتیکی مانند PCA باشد. در واقع آموزش Autoencoder با هدف کاهش خطای بازسازی است و آموزش آن جدا از مدل اصلی زمان بر است، در حالی که PCA یک تبدیل خطی می باشد که داده ها را در جهت ها با بیشترین واریانس تصویر می سازد.

در کل نتیجه می گیریم که کاهش بعد در مورد این دادگان خیلی تاثیر گذار نیست، دلیل آن می تواند این باشد که در تصاویر مختلف واریانس داده به صورت یکنواخت در همه ابعاد تصویر (پیکسل ها) قرار

گرفته اند و کاهش ابعاد تاثیر چشمگیری در افزایش دقت ندارد. برای دادگان تصویر شبکه های کانولوشنی عملکرد بهتری دارند.

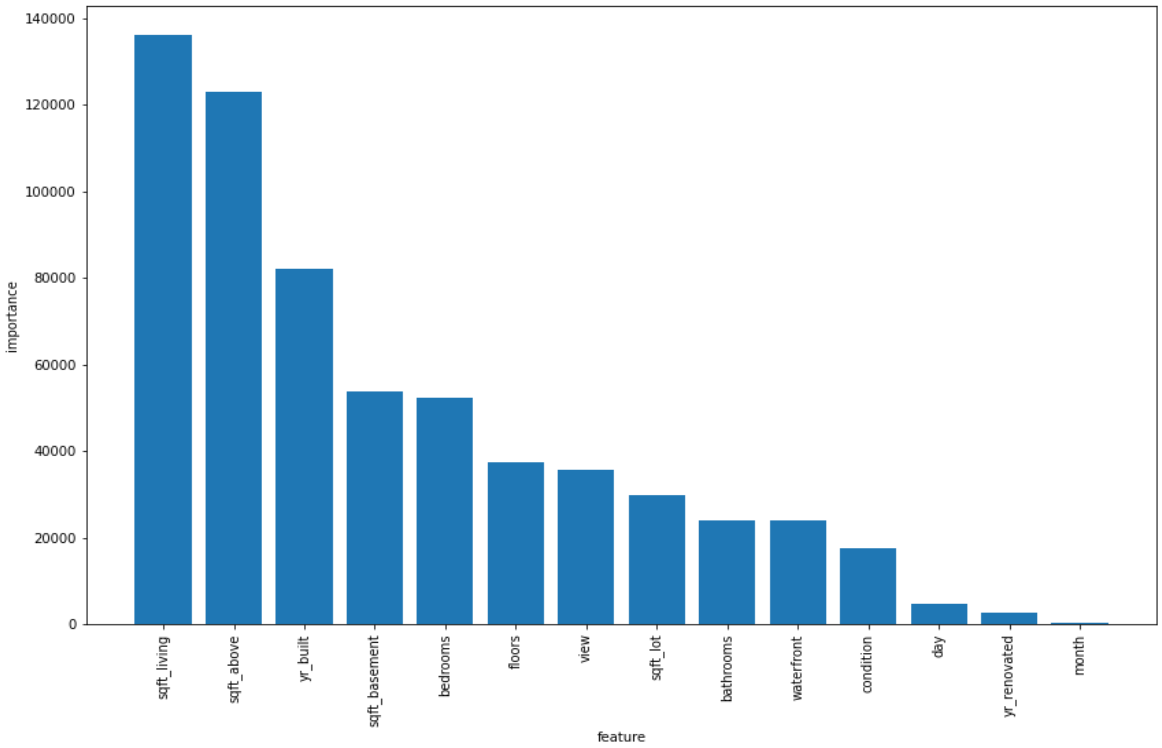
د)

ماتریس همبستگی برای داده های عددی را رسم می کنیم

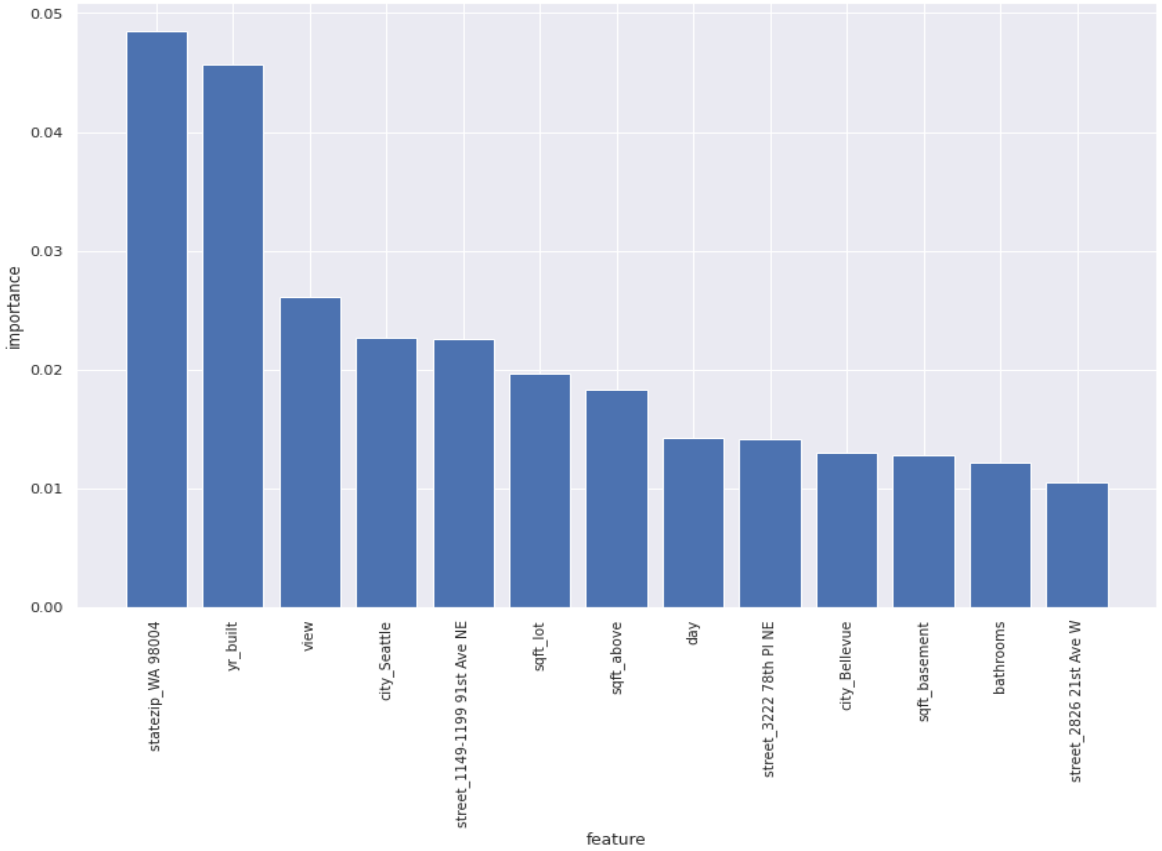


شکل ۳-۱۰ ماتریس همبستگی برای ویژگی های دادگان قیمت خانه

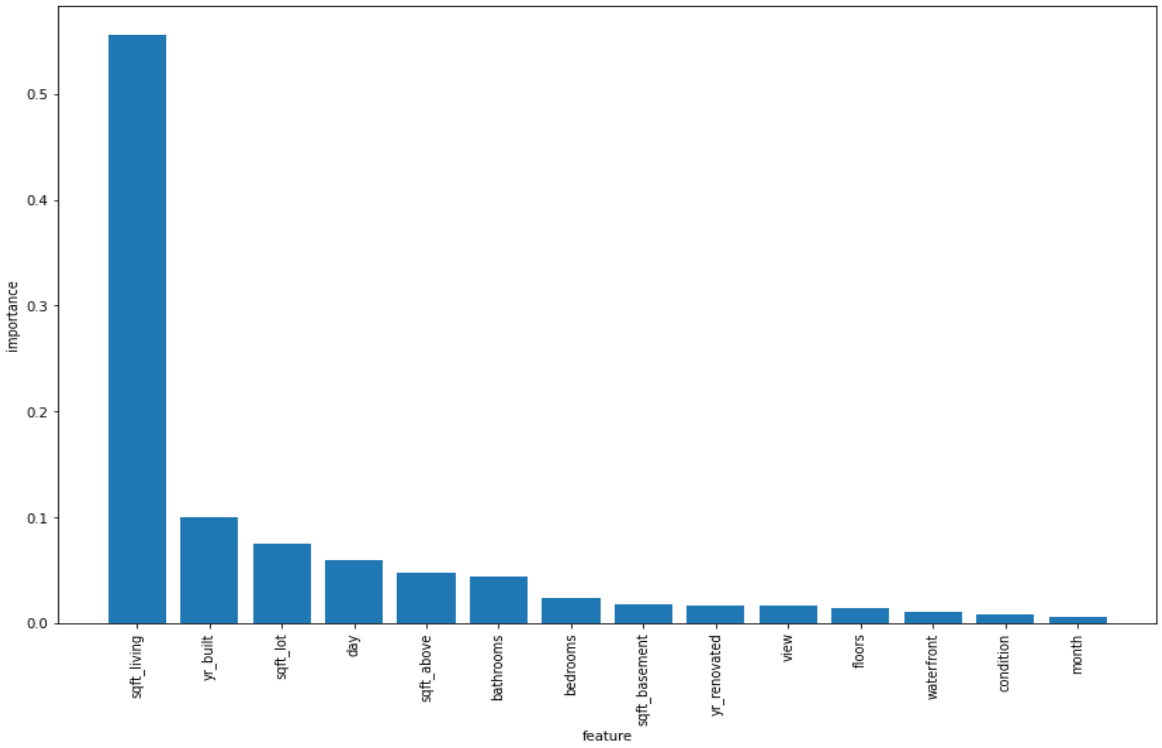
هر چه اعداد طلاقی دو ویژگی مثبت تر باشند به معنی آن است که افزایش یکی افزایش باعث افزایش دیگری می شود، و اگر این عدد منفی تر باشد به معنی این است که با یکدیگر رابطه عکس دارند و افزایش یکی باعث کاهش دیگری می شود. مثلا برای پیش بینی کردن قیمت خانه می توانیم نگاه کنیم کدام خانه ها در ردیف قیمت به ۱ یا ۱- نزدیک تر هستند، مثلا می بینیم که مساحت هال نقش موثری در افزایش ارزش خانه دارد.



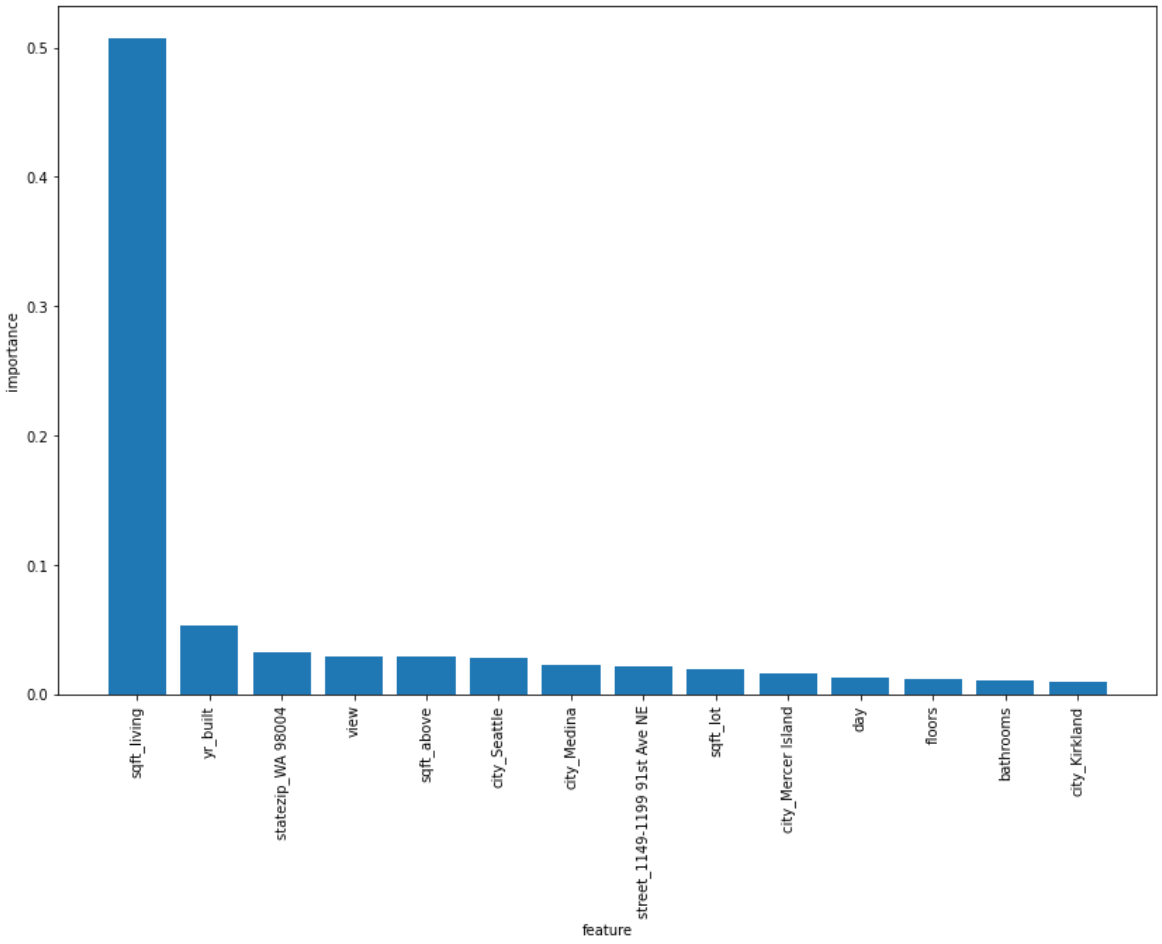
شکل ۳-۱۱ اهمیت هر یک از ویژگی های دادگان بر حسب روش رگرسیون خطی(فقط ویژگی های عددی)



شکل ۳-۱۲ اهمیت هر یک از ویژگی های دادگان بر حسب روش رگرسیون خطی(ویژگی های عددی به همراه غیر عددی)

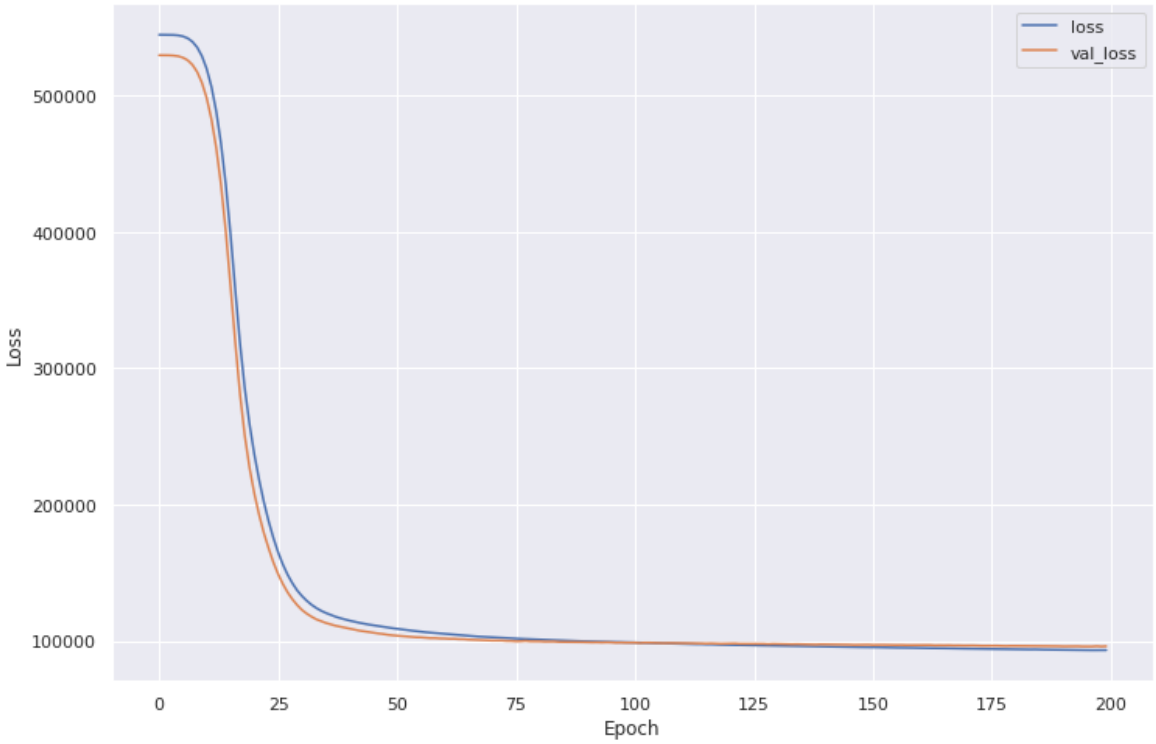


شکل ۳-۱۳ اهمیت هر یک از ویژگی های دادگان بر حسب روش درخت تصمیم(فقط ویژگی های عددی)



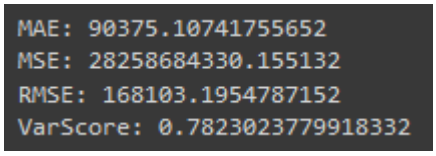
شکل ۳-۱۴ اهمیت هر یک از ویژگی های دادگان بر حسب روش درخت تصمیم(ویژگی های عددی به همراه غیر عددی)

حال شبکه سوال ۲ را تنها با ۱۰۰ ویژگی برتر(کاهش ابعاد) به دست آمده آموزش می دهیم



شکل ۳-۱۵ خطا بر حسب ایپاک برای داده های آموزش و ارزیابی

شکل ۳-۱۶ داده های پیش بینی شده به رنگ آبی در مقابل قیمت واقعی روی خط قرمز رنگ



شکل ۳-۱۷ معیار های مختلف برای داده های تست

می بینیم که تنها با ۱۰۰ بردار ویژگی به دست آمده از روش PCA شبکه به خوبی آموزش داده می شود و حتی نسبت به سوال ۲ افزایش دقت و کاهش خطا نیز داشتیم و زمان آموزش نیز به مراتب کمتر از سوال ۲ است. دلیل آن حذف ویژگی های وابسته به هم و بی فایده است، چون مثلا قیمت خانه ممکن است خیلی ربطی به ماه خرید آن نداشته باشد و این ویژگی های اضافی تنها باعث فرابرازش مدل می شوند.