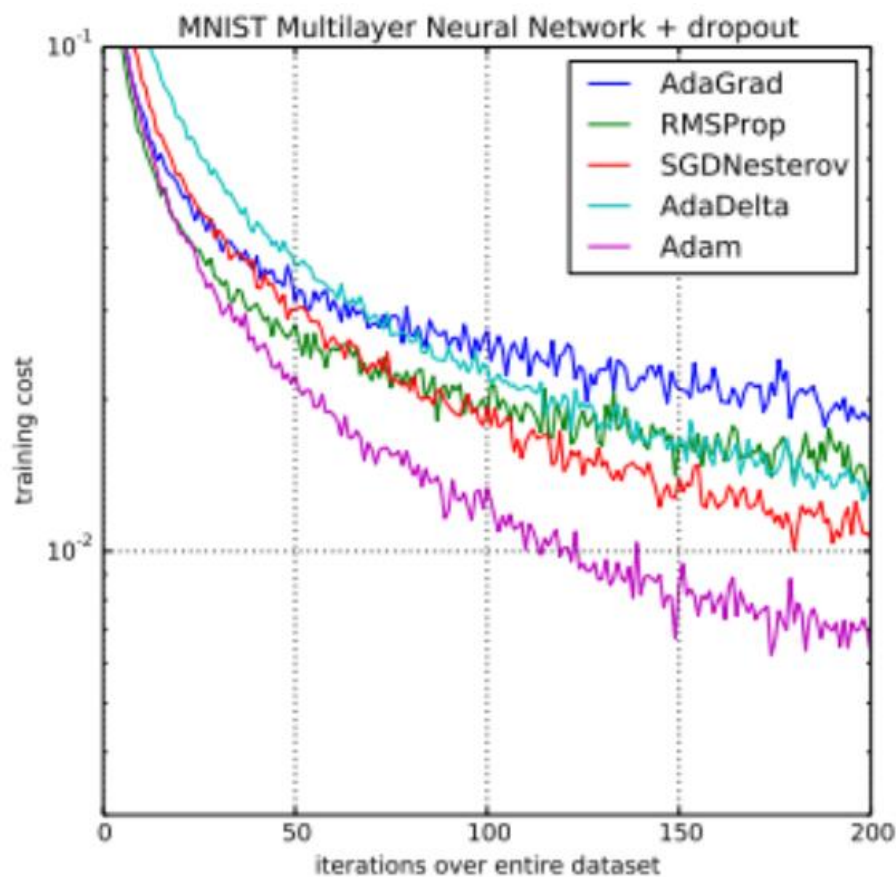


بخش 1:

(1)

مهم ترین نکته این است که بدانیم بر روی چه دیتای این optimizer ها را تست کرده ایم.



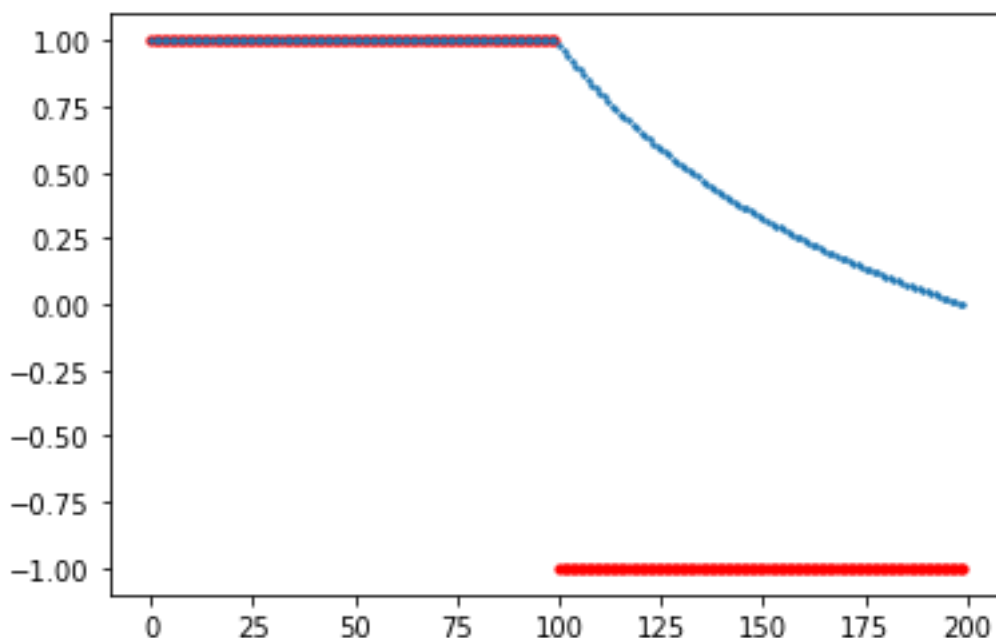
طبق این [لینک](#) الگوریتم adagrad زمانی تغییرات بیشتری در خلاف جهت شیب میدهد که شیب ها در ابعاد مختلف پراکندگی زیادی داشته باشند در غیر این صورت تغییرات در خلاف راستای شیب بسیار کند است همانطور که در نمودار می بینیم.

در الگوریتم RMSprop روند بهتر نسبت به adagrad شاهد هستیم چون نرخ کاهش ما وابسته به میانگین اندازه گرادیانت های گذشته ماست که، در مواردی که دیتاهای ما نویزی هستند بسیار پرکاربردتر است، زمانی که از sgdNestrov استفاده کردیم سرعت همگرایی بالا تری داشتیم به علت اینکه تقریباً 0.9 مقدار گرادیانت های قبلی در هر استپ به مقدار تغییر نرخ یادگیری کمک کردند و adam هم که اکثراً پیشنهاد می شود از ترکیب adagrad و RMSprop به خوبی استفاده کرده تا سریع تر همگرا شود.

(2)

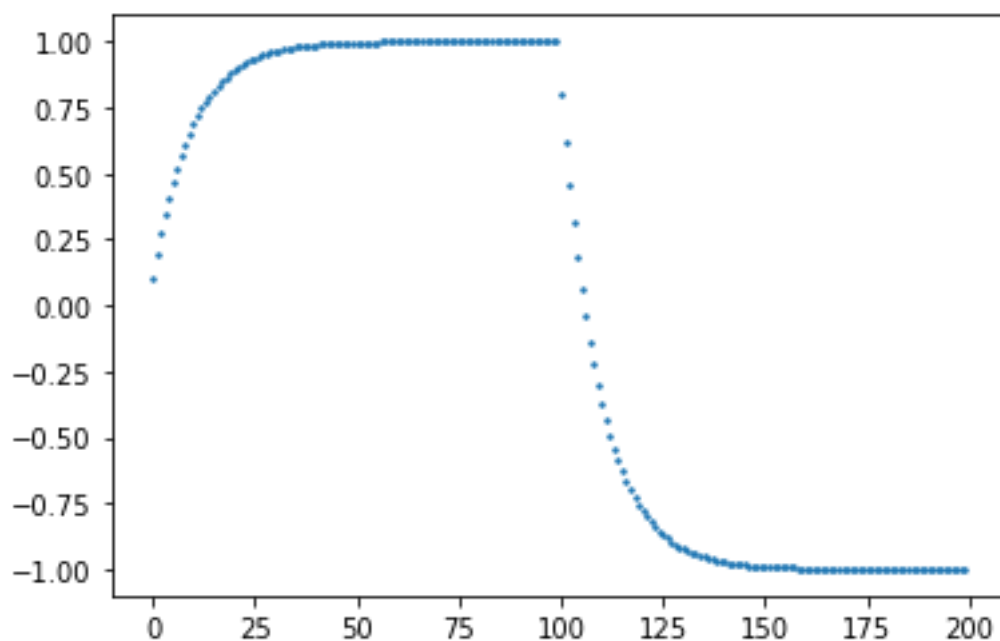
انتخاب optimizer بسیار به نوع دیتا وابسته است از لحاظ دقت، پراکندگی و خب هرچه دیتا بیشتر باشد برای هر بهینه ساز هم بهتر است و مورد بعدی مشاهده تغییرات گرادین ها در epoch های مختلف است که باعث می شود در تصمیم گیری ما برای انتخاب بهینه ساز موثر باشد. می توانیم زمان محاسبه هم اضافه کنیم که الگوریتمی مثل sgd+momentum رقابتی در این حوزه نمیتواند داشته باشد. اما بصورت کلی اکثراً adam را پیشنهاد می دهند در این [مقاله](#) کلاً به برتری adam اشاره کرده است.

بخش 2:



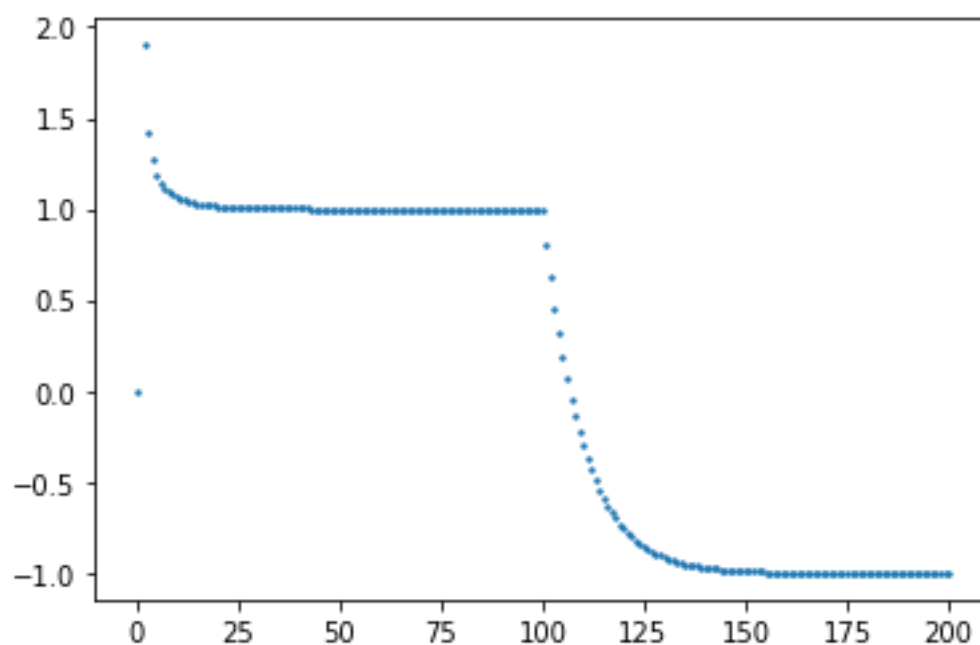
نقاط قرمز d هستند و نمودار ابی میانگین خواسته شده که در 100 تای اول چون مقدار متناسب با زمان افزایش یافته است تغییری نداشته و 1 مانده اما به تدریج از 100 به بعد به علت کاهش یک واحدی مقدار ان به صفر میل کرده است

(2)



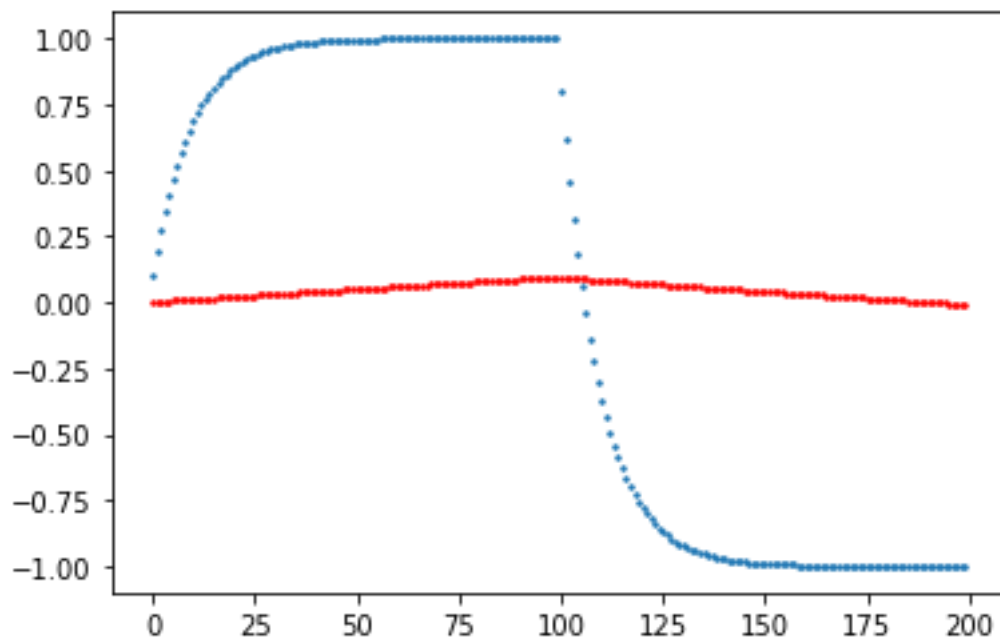
در این نمودار شاهد محاسبه $m1$ هستیم که مقادیر در زمان 25 به مقدار 1 رسید و همینطور در زمان 125 (25 زمانی که از تغییر مقادیر به 1- گذشته) نیز به 1- همگرا شده است که نشاندهنده آن است در این معادله روند میزان حجم تغییرات در صورتی که همپوشانی داشته باشند (تغییرات پیوسته باشد) سریع است

(3)

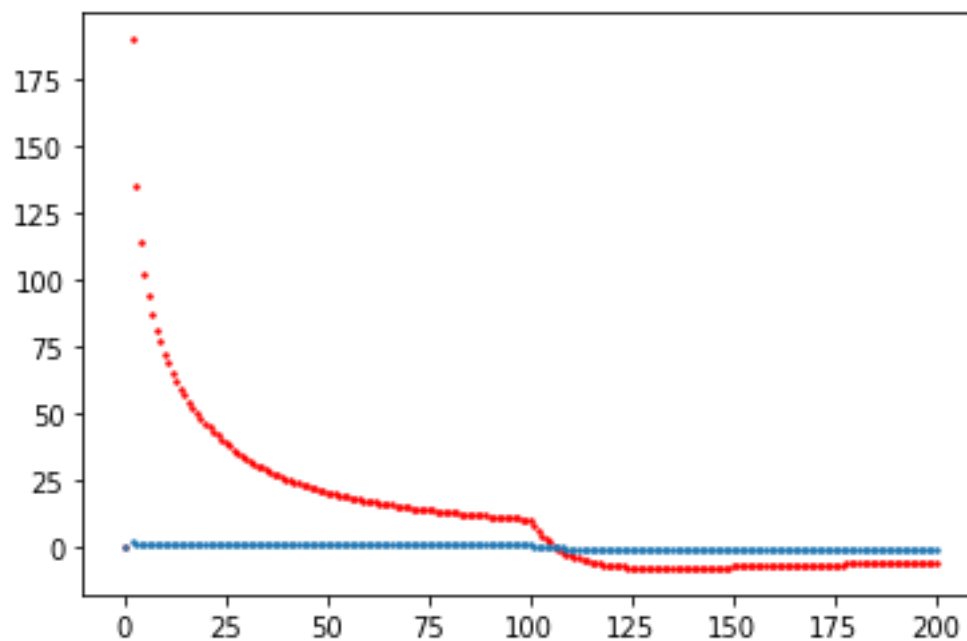


در این معادله با اعمال مقادیر معادله قبل روند تغییرات بسیار سریع تر انجام میشود

(4)



فرمول m_1 نمودار قرمز: $B=0.999$ و نمودار آبی $B=0.9$ با افزایش بتا شاهد کاهش میزان تغییرات بودیم به این دلیل که میزان تاثیر مقدارهای جدید را در محاسبه کمتر کرد

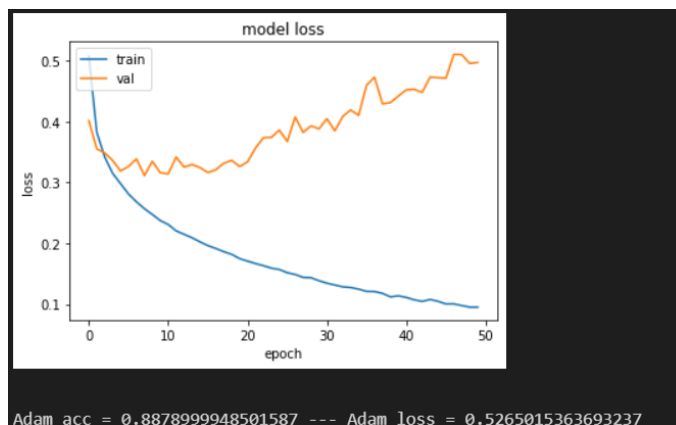
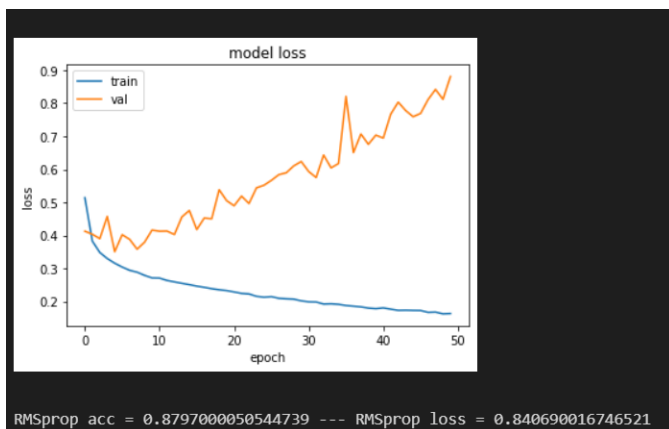
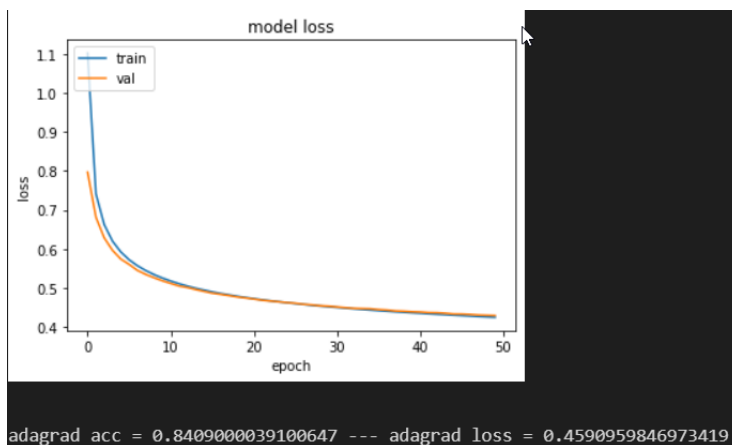
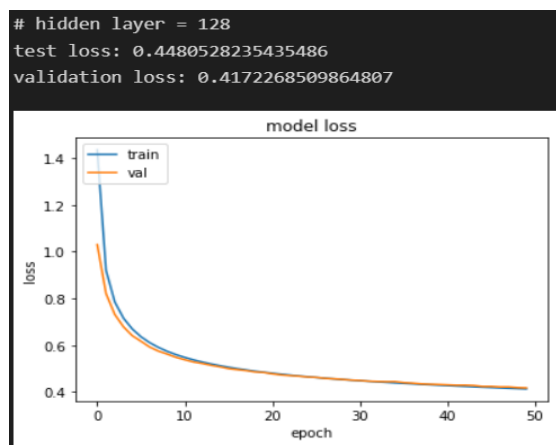


فرمول m_2 نمودار قرمز: $B=0.999$ و نمودار آبی $B=0.9$

طبق فرمول زمانی که بتا نزدیک به یک با شد در زمان ها بالا تاثیر مخرج کسر کم می شود و فرمول $m2$ همان $m1$ می شود اما در زمان های ابتدایی طبق تقسیم روند تغییرات را بسیار بیشتر می کند پس با نزدیک کردن بتا به یک این تغییرات بزرگتر می شود

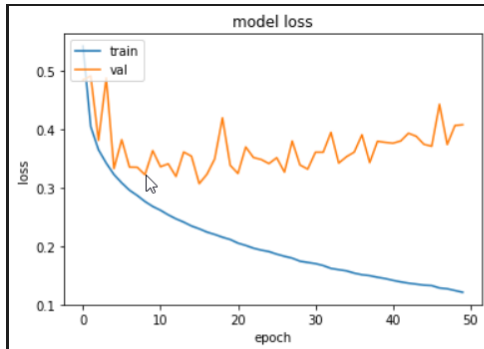
بخش 3:

- (1) مجموعه ای از دیتاهای پوشاک 10000 برای تست و 60000 برای آموزش
- (2) 10 درصد اما به نظرم زیر تا 30 درصد هم مشکلی ندارد طی آزمایش هایی دیگری که انجام دادم حتی تا 60 درصد تغییری خاصی مشاهده نشد و صرفا هزینه محاسباتی را افزایش داد
- (3) دقت فاز آموزش : 78 درصد دقت فاز ازمون: 77 درصد
از انجا که این دقت ها بهم نزدیک هستند این نتیجه را می شود برداشت کرد که مدل به خوبی آموزش دیده و overfit هم رخ نداده است
- (4) تغییراتی پیش امد با اضافه کردن دیتای اعتبارسنجی درصد خطا ها مقداری افزایش یافت اما تاثیری در انتخاب مدل بهتر نداشت درصد های انتخاب شده: 30، 10 و 60 و بهترین اندازه نرون ها 128 انتخاب شد
- (5)

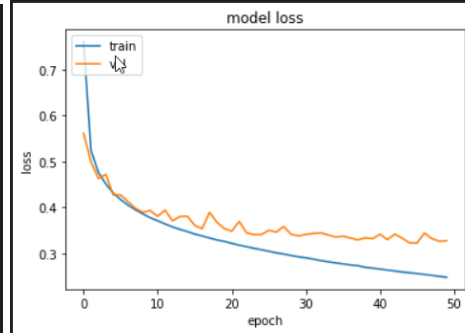


Adam و RMSprop دقت خوبی در داده های تست داشتند اما به با توجه به نمودار که داده اعتبار سنجی واگرا شده است به نظر مدل به خوبی آموزش ندیده است و adagrad نتیجه بهتری داشته و loss در SGD کمتر از بقیه بوده است

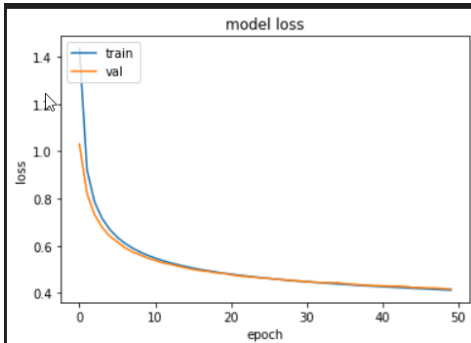
(6)



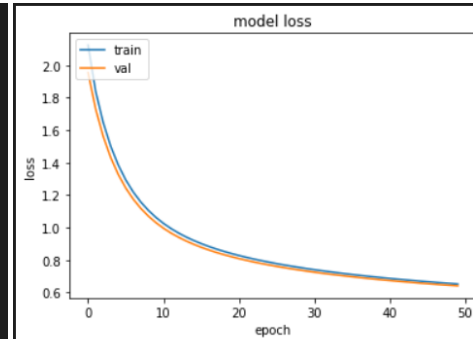
lr: 0.1 - test_loss: 0.4240674674510956 - test_acc = 0.4240674674510956



lr: 0.01 - test_loss: 0.35005438327789307 - test_acc = 0.35005438327789307



lr: 0.001 - test_loss: 0.4480528235435486 - test_acc = 0.4480528235435486



lr: 0.0001 - test_loss: 0.67023766040802 - test_acc = 0.67023766040802

در 0.1 مدل خیلی سریع تر برای داده های آموزش همگرا شد اما دقت اعتبار سنجی پایین است که به معنای overfit شدن است

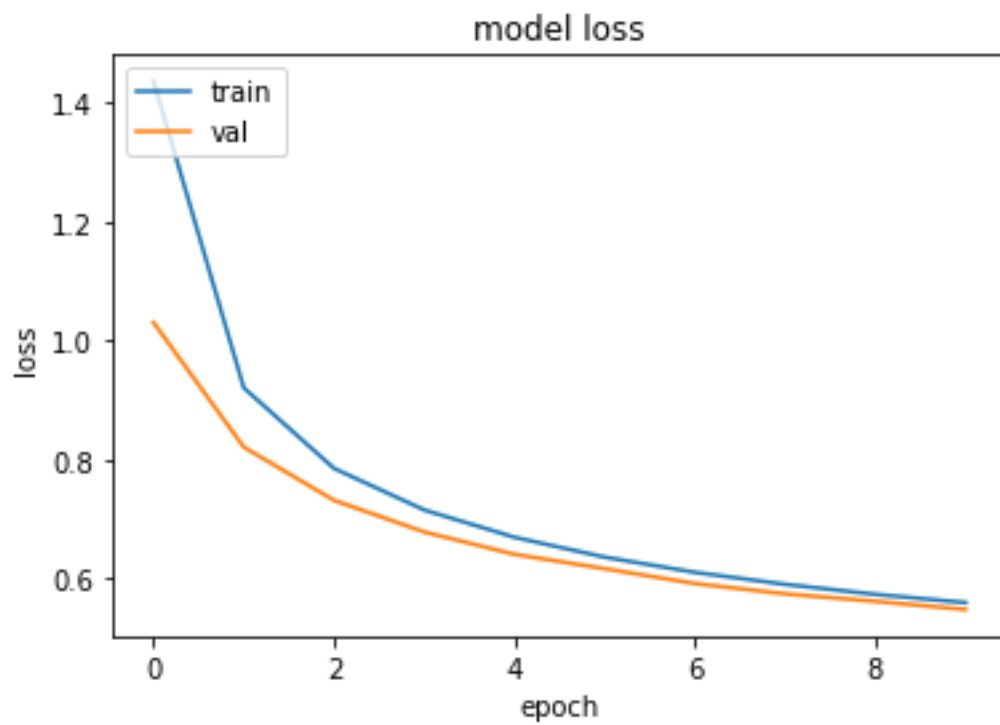
در 0.01 دیرتر همگرا شد اما دقت داده اعتبار سنج بالا تر رفت

در 0.001 داده های اعتبار سنجی و آموزش بر روی هم منطبق هستند ب

در 0.0001 مدل بسیار دیر تر همگرا می شود

(7)

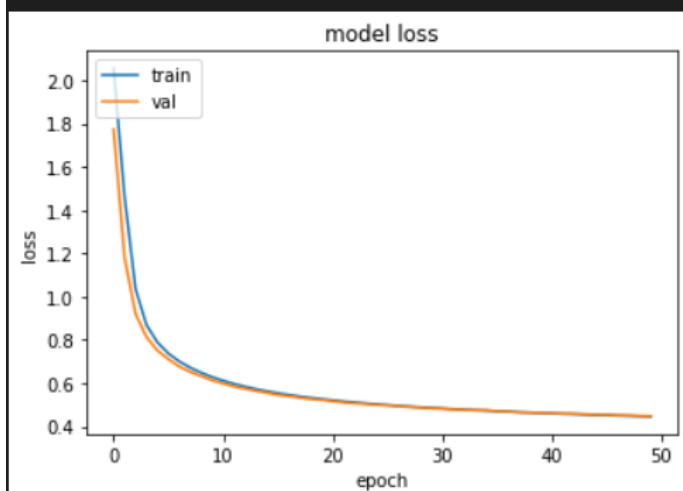
خب طبق انتظار نسبت به epoch 50 همگرایی کمتری دارد



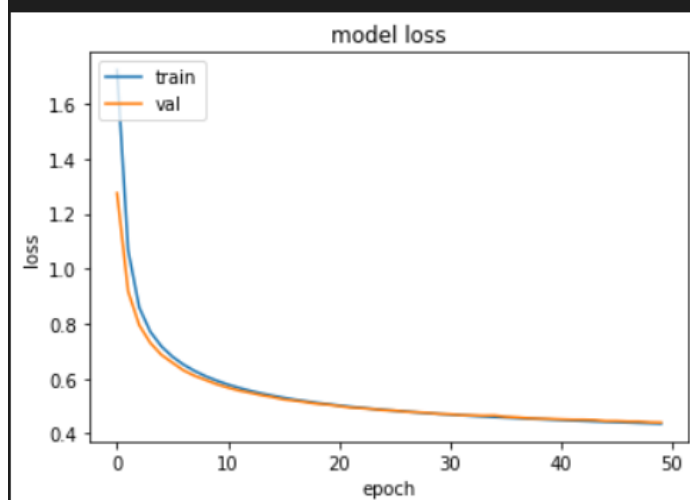
(8)

خطای اعتبارسنجی و آموزش برای تعداد نرون های مختلف

```
# hidden layer = 16
test loss: 0.47793740034103394
validation loss: 0.44793590903282166
```



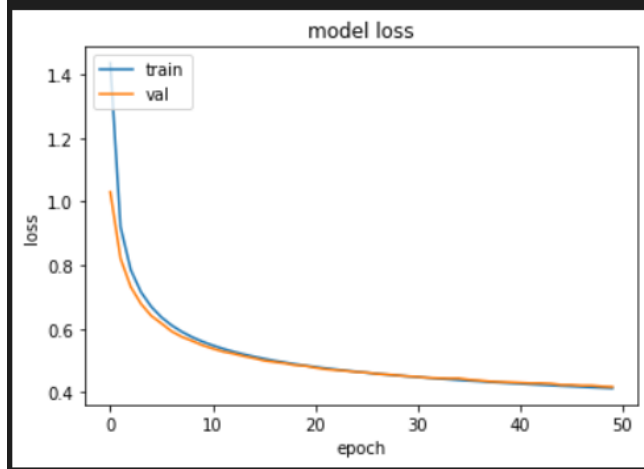
```
# hidden layer = 32
test loss: 0.47026968002319336
validation loss: 0.4398651123046875
```



hidden layer = 128

test loss: 0.4480528235435486

validation loss: 0.4172268509864807



hidden layer = 64

test loss: 0.4568256735801697

validation loss: 0.4270664155483246

