سید حامد سادات **400521405**

بخش 1)

در این سوال یک نمونه از چگونگی تولید داده‌های مصنوعی بر اساس توابع ریاضی سفارشی و آموزش یک مدل شبکه عصبی برای پیش‌بینی خروجی این توابع است از کتاب خانه های روبرو برای این بخش استفاده شده numpy ، matplotlib، kerasو sklearn استفاده شده.

بخش data generator:

این بخش مسئول تولید توابع مورد نیاز است
توانایی تولید توابع از ساده تا پیچیده و تعیین میزان نویر و تایین بازه تغییرات تابع با استفاده از توابع مختلف و تعیین تعداد نقاط خروجی را دارد

با دادن توابع مورد نظر،عرض از مبدا اولیه و ضرایب توابع مورد نظر به صورت رندوم تعیین شده و در دامنه تعیین شده نقاطی رندوم از تابع را در اختیار قرار میدهد

تابع قابلیت scale شدن در در نقاط بین 0 تا 1 را دارد

در حالت عادی از توزیع همانی استفاده میشود اما میتوان از توزیع نرمال نیز استفاده کرد اما پارامتر های loc و scale باید دستی تنظیم شود

بخش مدل:

مدل 5 لایه پنهان با تابع فعالساز relu دارد

و برای تابع loss از mse استفاده شده و برای خروجی تنها یک node هست که مقدار آن،مقدار پیشبینی شده تابع توسط مدل خواهد بود

نکات درباره مدل:

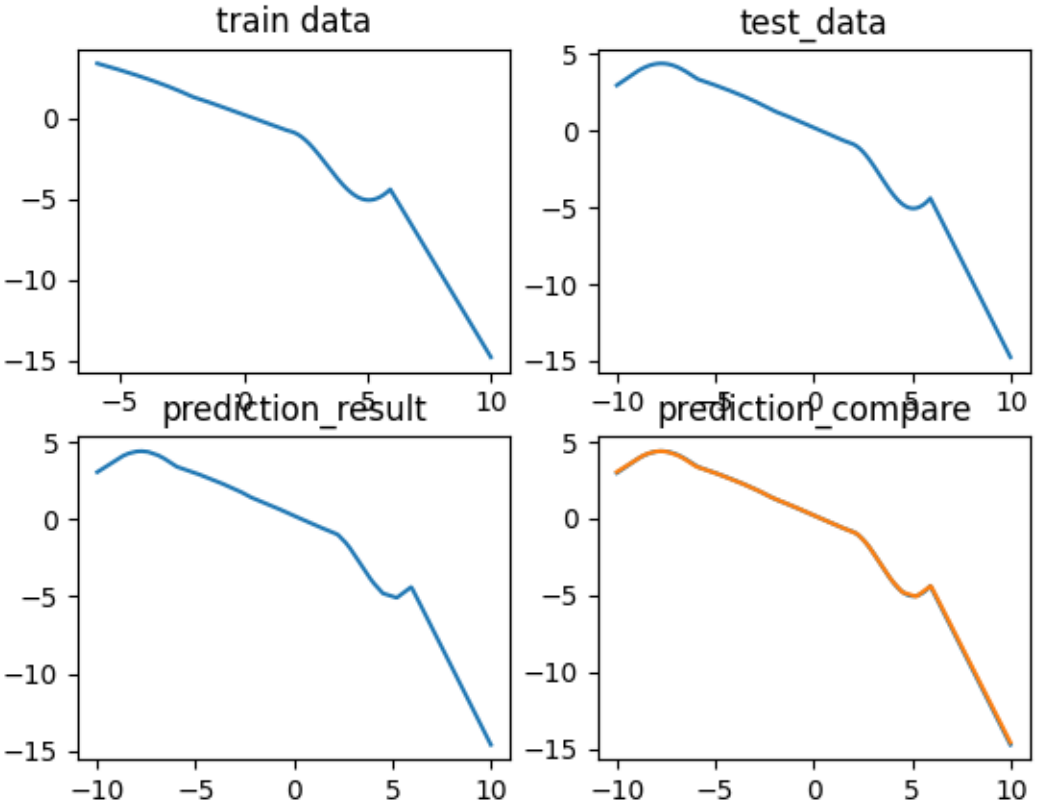
میتوان برای تواب ساده با تعداد کمتری لایه خروجی دقیقی گرفت اما از آنجایی که این مدل باید برای توابع پیچیده هم پاسخگو باشه،تعداد لایه هارو زیاد کردم

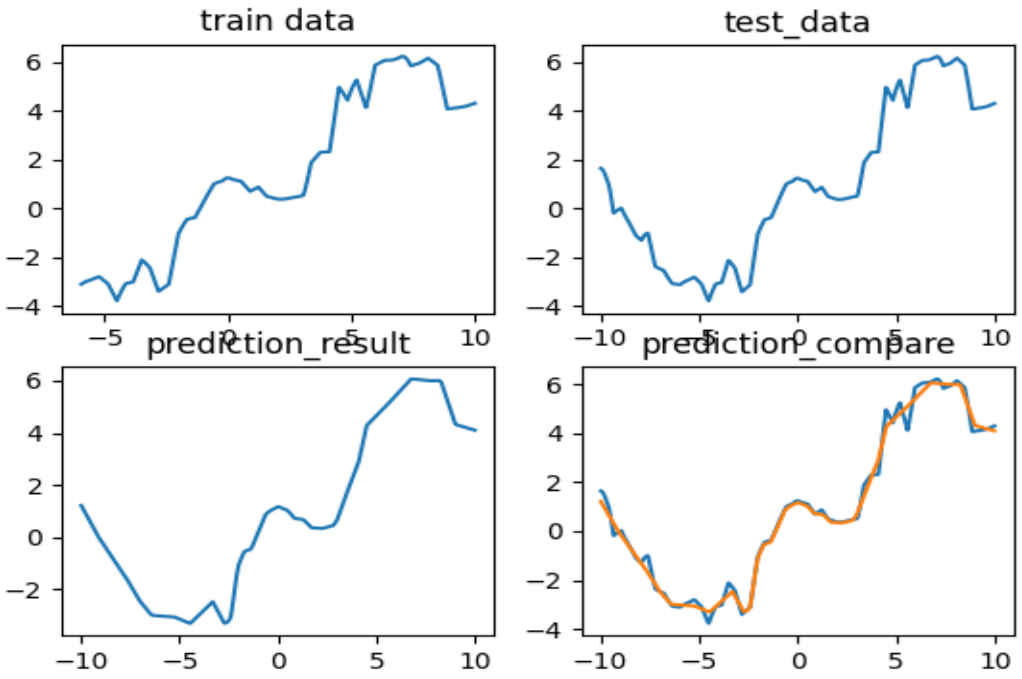
این مدل در توابع سینوسی کمی مشکل دارد اما با بالا بردن تعداد لایه ها یا کم کردن اندازه batch(ترجیحا batch_size=1) ها این مشکل رفع میشود،البته این مشکل با فرکانس تابع مثلثاتی رابطه مستقیم داشت

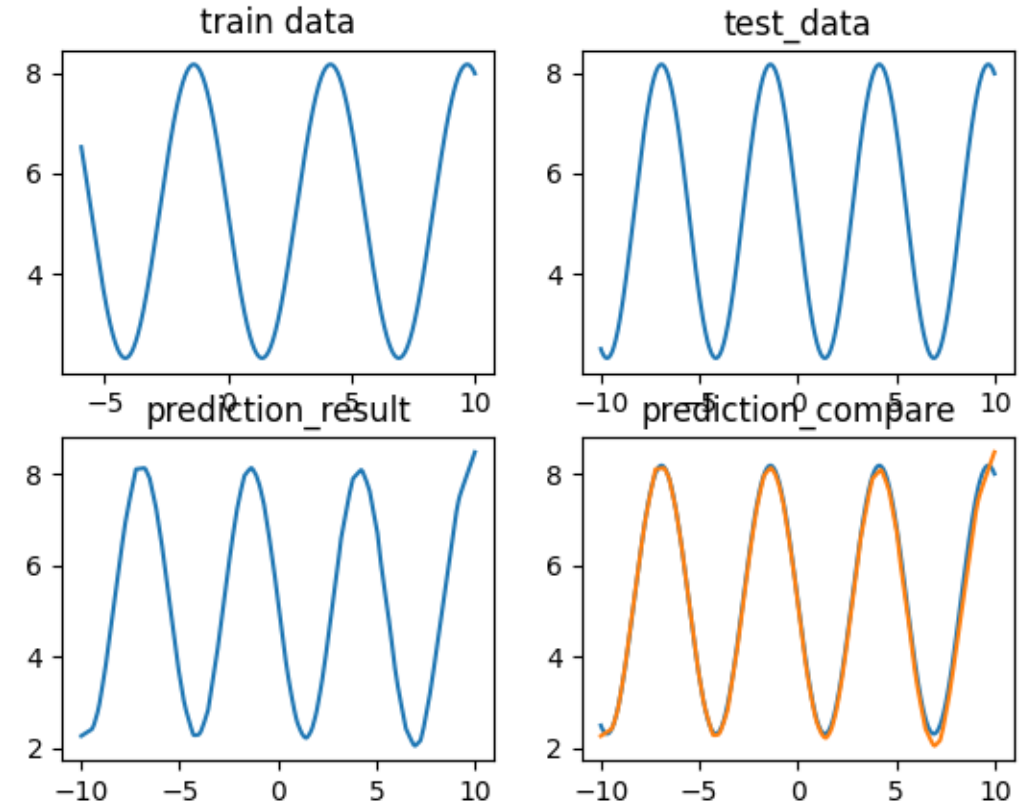
به طور کلی دقت تابع با افزایش تعداد لایه ها،تعداد داده،تعداد epoch دقت مدل افضایش پیدا میکرد

با کاهش تعدادbatch ها دقت تابع افضایش خوبی داشت اما در به نویز ها حساستر میشد

عکس هایی از تعدادی از مثال ها در پوشه asset وجود دارد




train data
test_data
prediction_result
prediction_compare


train data
test_data
prediction_result
prediction_compare

بخش 2)

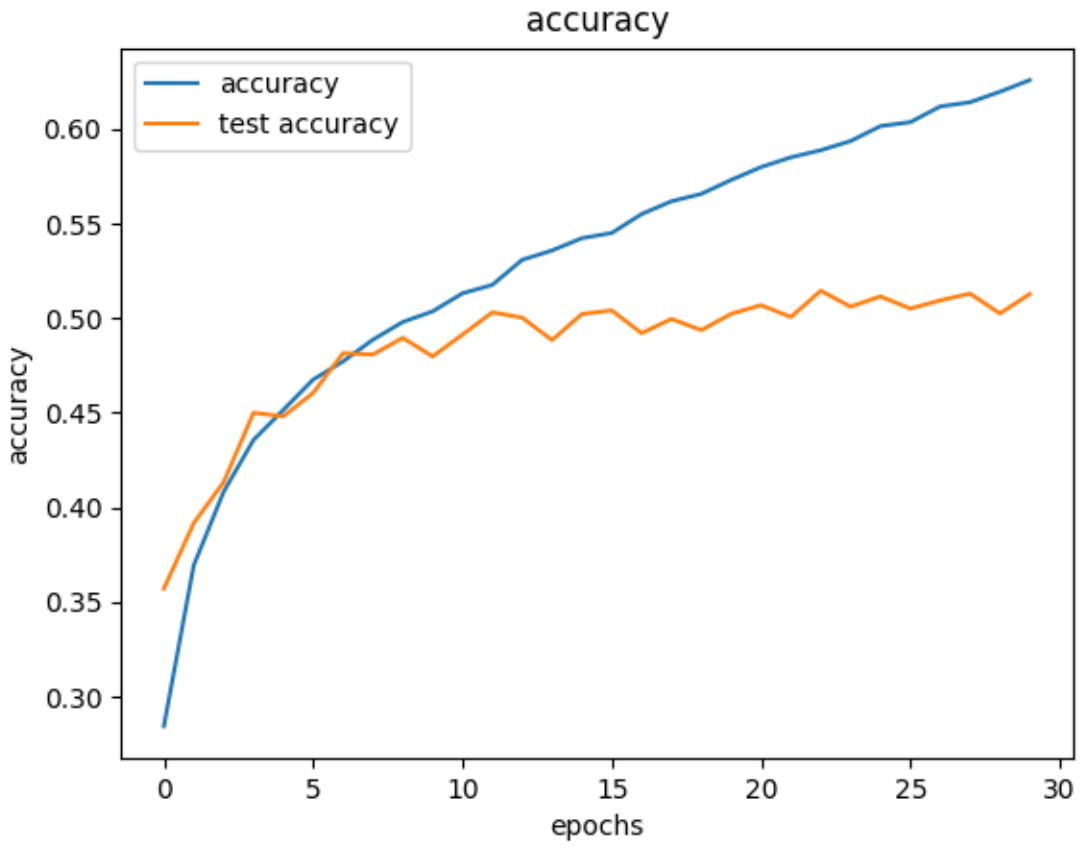
بعد از load کردن دیتای cifar-10 و آماده کردن آن برای train

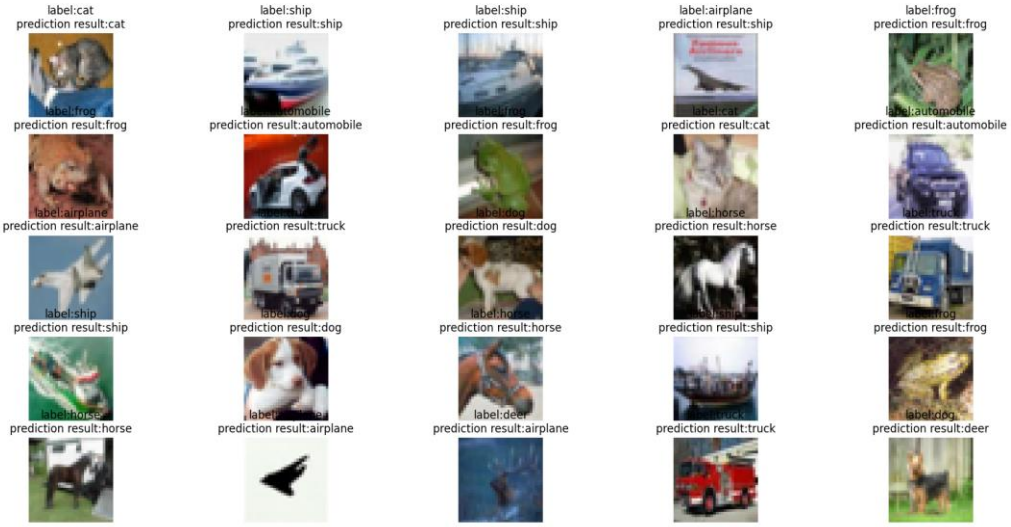
این اطلاعات را به تابع برای train میدهیم

مدل بعد از train شدن میتواندsave شود و در صورت نیاز load شود

درباره مدل:

مدل از 6 لایه پنهان dense تشکیل شده که دقت تقریبا 51 و تست 60 میدهد گرچه اگر از لایه های convolutional و maxpooling و dropout و BatchNormalization() استفاده کنیم،دقت مدل در تست به 86 درصد هم میرسد





بخش3)

دیتا با کمک فایل csv لود شده و دیتا ست به 20 درصد برای تست و 80 درصد برای آموزش تقسیم شده

درباره مدل:

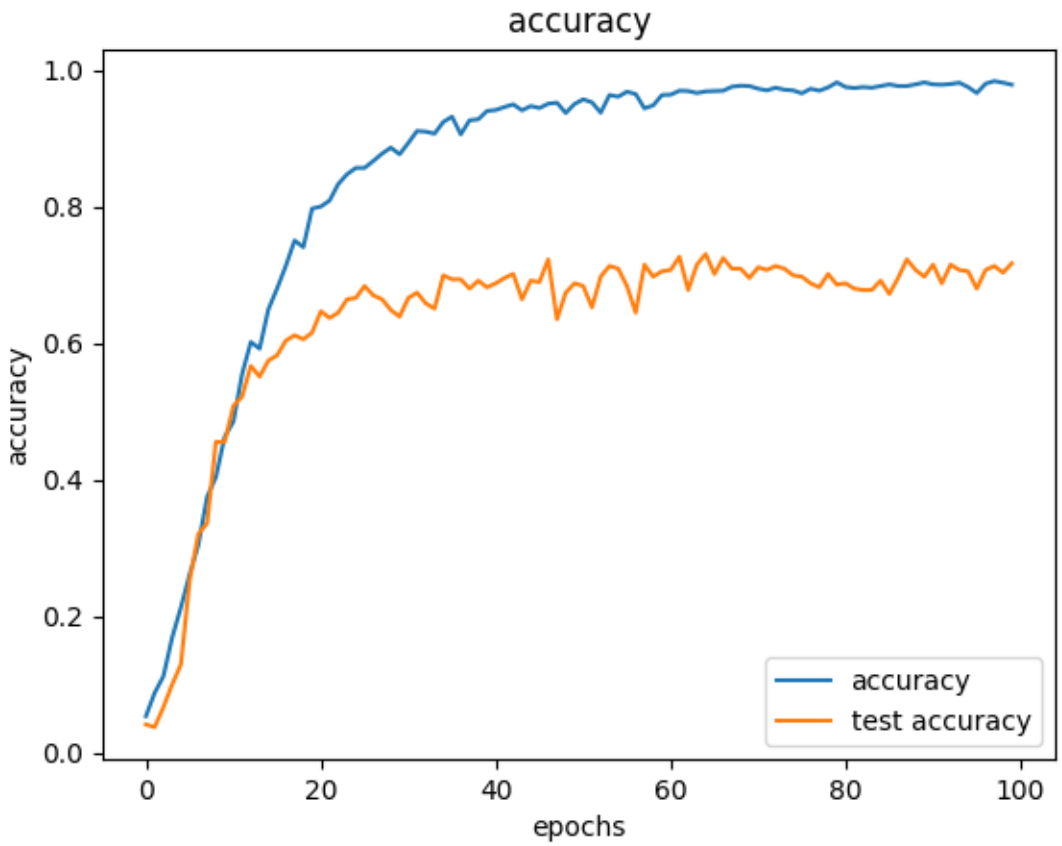
اگر فقط از dense استفاده کنیم،به 50 درصد دقت برای تست و 80 درصد دقت برای آموزش میرسیم که نشانه اورفیت شدن مدل است

برای درست کردن مشکل دقت train از دو لایه conv و BatchNormalization

و یک لایه dropout و maxpooling استفاده کردیم و در نتیجه دقت آموزش به 98 و دقت تست به 70 درصد رسید

گرچه مشکل اورفیت کردن درست نشد اما حداقل شاهد افزایش دقت مدل بودیم

مدل فقط برای تشخیص چهره استفاده شده


accuracy
1.0
0.8
0.6
0.4
0.2
0.0
accuracy
0
20
40
60
80
100
epochs
accuracy
test accuracy