به نام خدا

پروژه درس بهینه سازی

دانشکده علوم ریاضی

دانشجو: احمد یزدانی و محمد جواد عباسپور

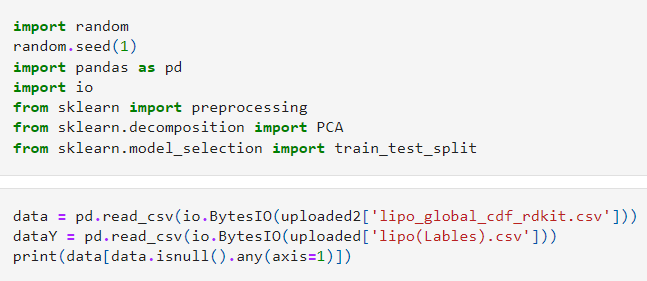
استاد راهنما: خانم دکتر طاهری

استاد درس: آقای دکتر بیژن احمدی

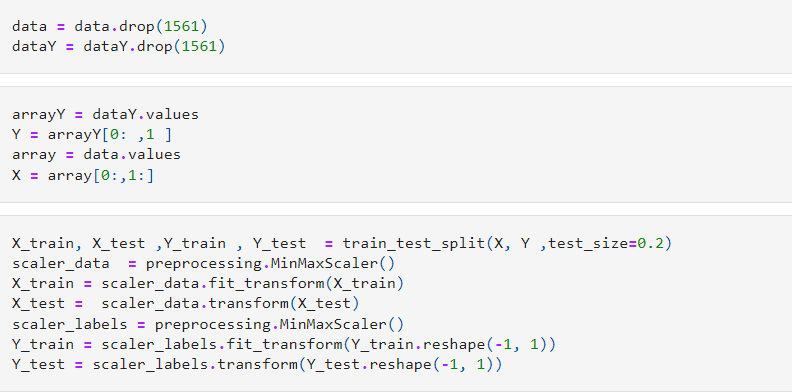
بهینه سازی هایپر پارامتر ها

هدف ما از انجام این پروژه بهینه سازی هایپر پارامتر های مدل عصبی بر روی دادهای lipo است.

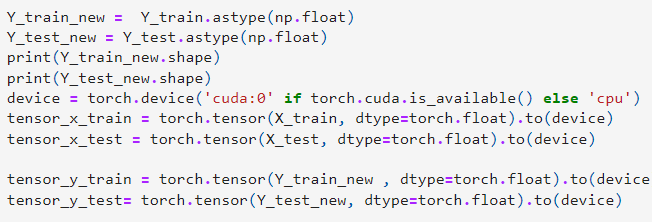
ابتدا طبق دستور زیر متغیر های پیش گو و متغیر هدف را فراخوانی میکنیم و ثبت هایی که گمشده هستند را میابیم:

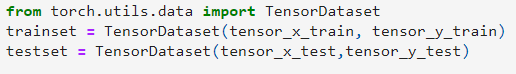


با خروجی گرفتن از کد فوق مشاهده میشود که ثبت 1561 باید حذف شود در نتیجه این ثبت را با دستور زیر حذف مینماییم و همچنین مجموعه هایمان را افراز مینماییم و همچنین به بازه ی صفر تا یک داده هایمان را انتقال میدهیم(زیرا این بازه بهترین خروجی را برای شبکه های عصبی دارد ):



حال آرایه هایمان را به فرمت torch تبدیل مینماییم و سپس به شکل مجموعه داده ی آموزشی و مجموعه داده ی تست طبق کد زیر در میاوریم:





حال تابعی به شکل زیر مینویسیم تا داده های آموزشی و داده های تست را به ما برگرداند:

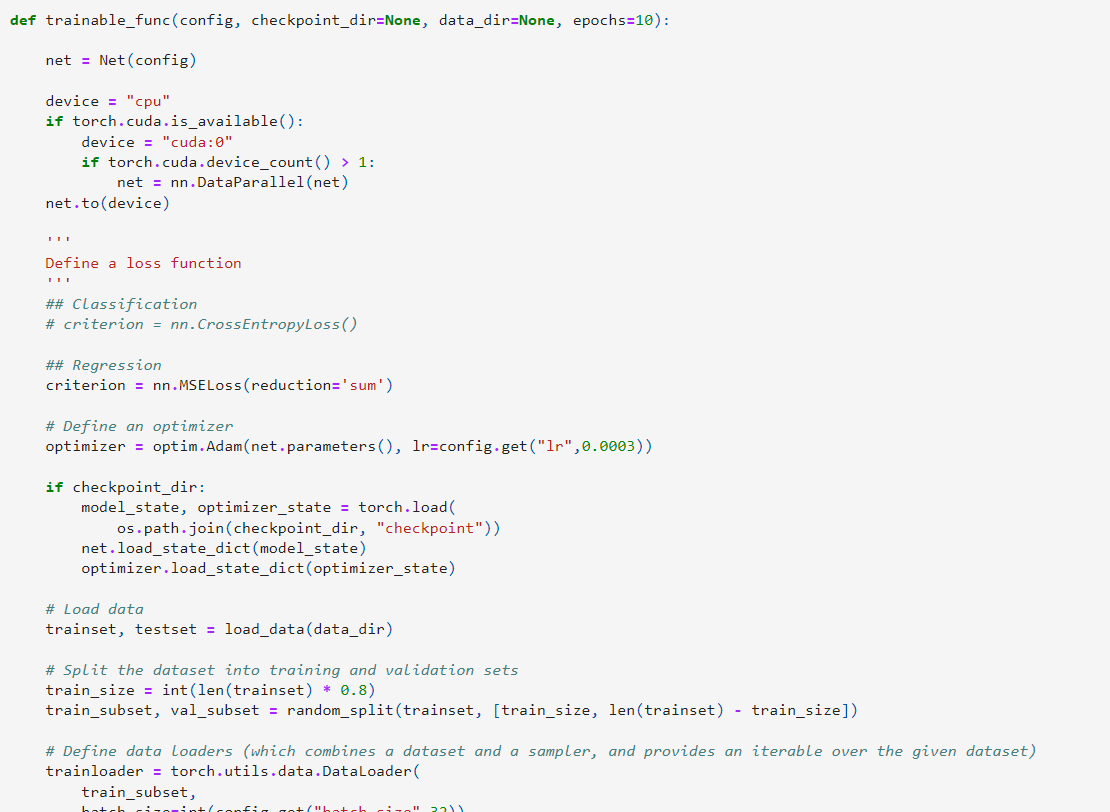


کلاس زیر را برای ساختن مدل عصبی با چهار لایه ی پنهان تعریف میکنیم که ابتدا بعد خروجی هر لایه را میگیریم(به صورت پیش فرض 100 است) سپس تابع فعال ساز هر لایه را به عنوان ورودی میگیریم(به صورت پیش فرض relu است) حال تابع activation\_function را برای فراخوانی ورودی های کاربر(برای مثال اگر ورودی tanh بود در پایتون به صورت torch.tanh باید فراخوانی شود) مینویسیم سپس عملیات هر لایه را نوشته و خروجی آخر که همان پیش گویی لیبل ها است را برمیگردانیم:

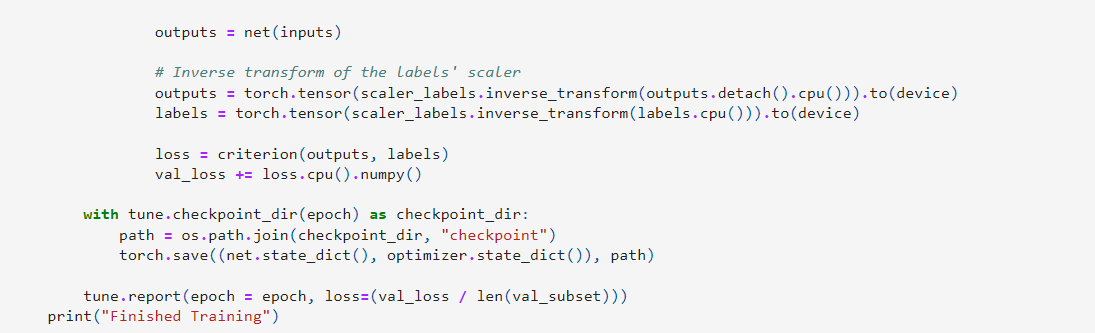


حال با تابع زیر مدل عصبی خود را آموزش میدهیم ابتدا یک آبجکت از کلاس عصبی خود تعریف کرده و سپس تابعی برای محاسبه خطا تعریف میکنیم حال نرخ آموزشی را با روش adam بهینه میکنیم(مقدار پیش فرض 0.003 و میتوان از SGD نیز استفاده نمود) حال این if زمانی true میشود که ray tune دستور ذخیره سازی مدل را بدهد.

حال در این قسمت شبکه ی عصبی خود را آموزش میدهیم و مقدار خطای آن را با تابع MSE محاسبه میکنیم:

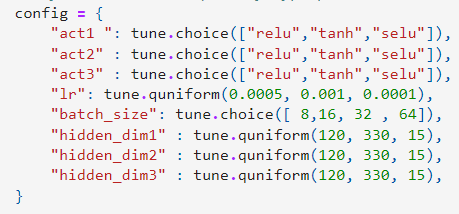






حال تابعی برای محاسبه دقت مدلمان مینویسیم که از MSE برای محسابه ی خطا استفاده میکند.

درآخر وارد تابع main میشویم که باتوجه به تست هایی که گرفته شد بهترین هایپر پارامتر های ورودی، هایپر پارامتر های زیر برای مدلمان است:

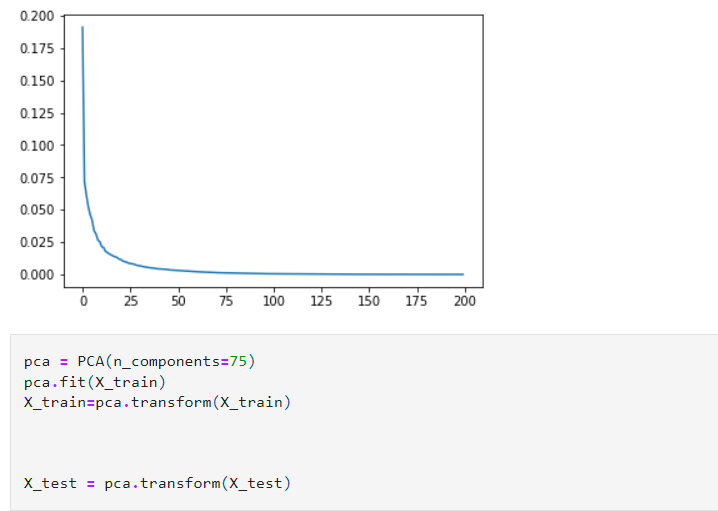


حال با استفاده از روش های مختلف برای محسابه ی بهترین هایپرپارامترها خروجی های نسبتا مختلف گرفته شد که در جدول زیر آورده شده است:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASHAscheduler  OptunaSearch | MedianStoppingRule  OptunaSearch | MedianStoppingRule  HyperOptSearch | ASHAscheduler  HyperOptSearch | search algorithm  and scheduler |
| 0.455 | 0.058 | 0.0619 | 0.459 | validation score |
| 0.454 | 0.458 | 0.552 | 0.488 | test set score |
|  | Overfit | Overfit |  |  |

همانطور که مشاهده میشود در سرچ هایی که از medianstop استفاده شده است باعث رخ دادن بیش برازش شده است.

حال با انجام دستور pca، مجدد بهترین مدل را انتخاب میکنیم پس طبق نمودار زیر مشاهده میشود 75 متغییر پیش نهاد شده توسط pca اطلاعات کافی متغییر ها را برای ما حفظ میکند پس 75 متغییر را انتخاب نموده و باقی را حذف مینماییم پس داریم:



حال با توجه به کاهش بعد انجام شده، مجدد با الگوریتم های سرچ مدل را انتخاب میکنیم:

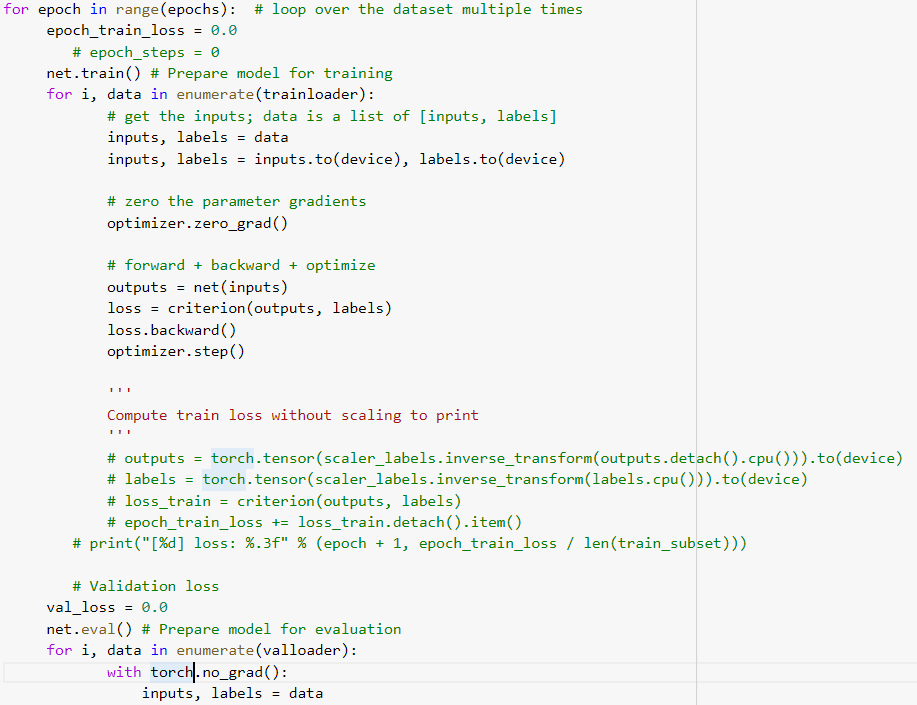
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ASHAscheduler  OptunaSearch | MedianStoppingRule  OptunaSearch | MedianStoppingRule  HyperOptSearch | ASHAscheduler  HyperOptSearch | search algorithm  and scheduler |
| 0.421 | 0.022 | 0.024 | 0.400 | validation score |
| 0.515 | 0.555 | 0.419 | 0.518 | test set score |
| Overfit | Overfit | Overfit | Overfit |  |

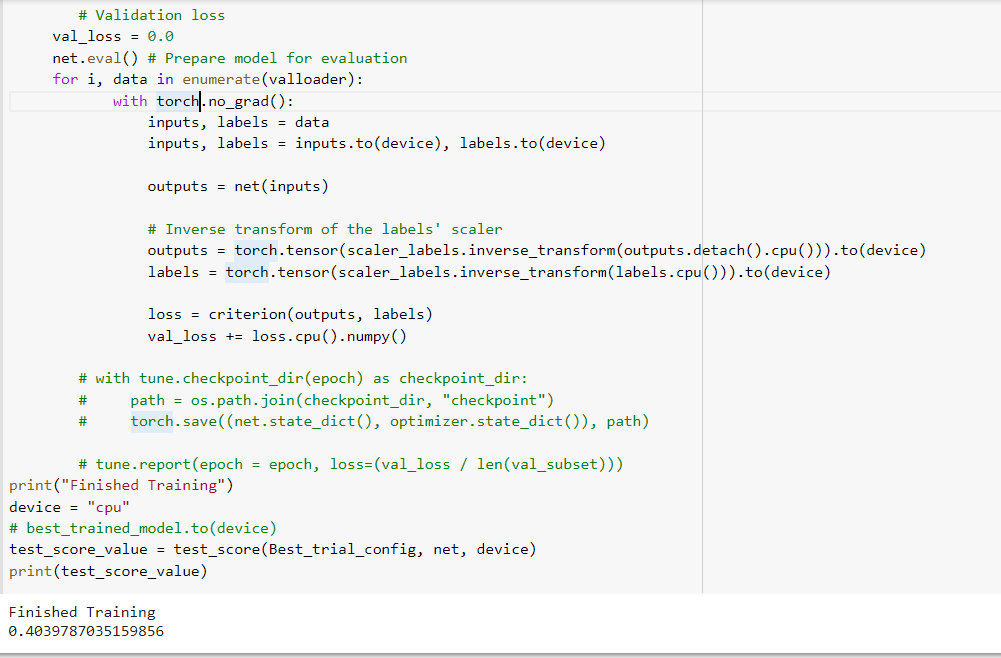
همانطور که مشاهده میشود با استفاده از pca تمام روش ها دچار بیش برازش شده است.

حال بهترین مدل با خطای 0.455 بر روی داده های آموزش و خطای 0.454 بر روی داده های تست است که هایپرپارامتر های آن با epoch = 50 برابر است با:



حال مدل خود را با هایپر پارامتر های فوق طبق کد زیر برازش میدهیم: 





همانظور که مشاهده میشود برای آموزش مدل بهترین پارامتر ها رو ورودی داده ایم و همچنین چون با epochs = 50 به بهترین دقت رسیدیم، برای برازش مدل نیز این epochs را برابر 50 قرار دادیم.

همانطور که مشاهده میشود دقت مدل برابر 0.403 شد.

با تشکر از توجه شما