Home Work No 3

Report

Yazdan Zandiye Vakili

610397193

1. **Pre-processing**

در این قسمت دیتاست را باز کرده و برچسب های داده های برچسب دار را جدا کردم و برای بخش پیش پردازش دو کار مطمئن شدن از عدم وجود NULL or NaN را انجام دادم، اما به صورت شهودی و چون دیتاست یک دیتاست بسیار مشهور بود و در کتاب خانه Keras به همین شهودی دیدن بسنده کردم. درباره نرمالایز کردن داده ها هم در ابتدا با استفاده از تابع zscore در کتاب خانه scipy این کار را انجام داده بودم ولی به دلیل اینکه در ادامه روال نیاز به اعداد صحیح داشتیم و این رند شدن خیلی ضرر به روند کار می زد و همچنین چون کل اعداد در بازه 0 تا 255 بودند، پس نیازی در واقع به این کار نبود.

1. **Build the Auto Encoder**

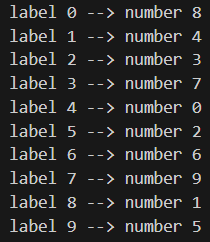
در این قسمت با استفاده از کتاب خانه TensorFlow یک Auto Encoder با سه لایه مخفی، به ترتیب 128 ، 16و 128 ساختم تا در نهایت بتوانیم داده ها را ابعادشان را از 728 به 16 برسانم. برای Compile کردن Auto Encoder از تابع SGD به عنوان Optimizer و از تابع MeanSquareError (MSE) به عنوان Loss Function استفاده کردم که نتیجه بهتری به من در طی روند داد (البته نتیجه کلی خوب نبود و به دلیل بودن تمرین در بازه امتحانات وقت برای بهینه تر کردن تمرین پیدا نکردم) و برای استفاده از Encoder ، دو ساختار Encoder و Decoder را جداگانه ساختم و در کنار هم قرار دادم و با هم Auto Encoder را پیاده سازی کردم و بر روی داده های آموزش برچسب دار، Fit کردم و بعد برای کمتر شدن زمان اجرا این Encoding را روی داده های تست برچسب دار و همچنین داده های آموزش بدون برچسب هم پیاده سازی کردم تا در روند Clustering به زمان کمتری برای اجرا نیاز داشته باشم (ولی خب همانطور که می بینیم دقت پایین آمد)

1. **Clustring (K-Means)**

برای خوشه بندی داده های بدون برچسب از الگوریتم K-Means استفاده کردم چون دقیقا به 10 برچسب نیاز داشتم و با الگوریتم DBSCAN به Post-processing هم برای رساندن تعداد خوشه ها به این عدد نیاز داشتم ولی با این الگوریتم خیر (هر چند که ممکن بود دقت بیشتری بدهد)

1. **Post-processing on Clustering**

حال همانطور که گفته شده بود در ویدیو توضیح پروژه در گروه، به صورت شهودی تعداد نمونه از هر خوشه برداشتم و با نگاه کردن گرفتن اکثریت عدد متناظر با هر برچسب را تعیین کردم و برچسب هر خوشه را به عدد متناظرش تغییر دادم. (در پوشه Sample\_Images ، 100 نمونه ای که گرفته بودم و با استفاده از کتاب خانه Pillow از روی ماتریس موجود بازسازی کرده بودم را می توانید مشاهده کنید ولی، همانطور که در خود پروژه هم ذکر کردم این نمونه 100 تایی - 10 تا از هر خوشه – نتوانست کامل باشد و مجبور شدم تعداد بیشتر نمونه بگیرم و نگاه کنم ولی برای ارسال فایل فقط همان 100 نمونه را ارسال کردم. همچنین یکبار برای مشخص کردن عدد متناظر به هر خوشه، مرکز هر خوشه را گرفتم و با استفاده از Decoder به ابعاد 784 بردم و در نهایت تصویر را بازسازی کردم ولی به دلیل دقت پایین Auto Encoder نتیجه اصلا رضایت بخش نبود و نمی شد به آن اتکا کرد و با توجه به این موضوع، همان نمونه هارا بیشتر کردم و به بررسی آن ها پرداختم و برای کمتر نشدن دقت از دیتاست ای از داده های بدون برچسب که همچنان آن را حفظ کرده بودم استفاده کردم و برحسب آن تضاویر را بازسازی کردم)  
در خود پروژه اعداد متناظر با هر خوشه را ذکر کردم و در پایین تصویر آن را می گذارم.

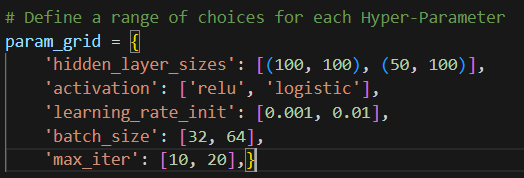


1. **Combining DataSets**

در این قسمت برچسب های بدست آمده را بعد از تغییر به عدد متناظرشان به دیتاست برچسب های موجود از داده های آموزش اضافه کردم و یک دیتاست از برچسب های داده های آموزش ساختم و همین کار را برای داده های آموزش انجام دادم و در نهایت با ترکیب داده های آموزش برچسب دار و بدون برچسب، یک مجموعه داده آموزش را ساختم.

1. Implement MLP Classifier with GridSearch

با استفاده از کتاب خانه Sci-kit Learn یک MLP Classifier ساختم و با استفاده از روش  
 5 Folds Cross-Validation این Classifier را بر روی دیتاست های ترکیب شده Fit کردم و با دادن مجموعه ای از پارامتر ها به این Classifier ، بهترین پارامتر ها را بدست آوردم. در پایین می توانید مجموعه پارامتر هایی که برای MLP Classifier در نظر گرفته شده بود را مشاهده کنید.



بهترین پارامتر ها در نهایت به این شکل شدند :

1. Hidden layer sizes : (100, 100)
2. Activation Function : Relu
3. Learning Rate Init : 0.001
4. Batch Size : 64
5. Max Iteration : 20

و در نهایت نتایج آموزش این Classifier بر روی این دیتاست ترکیب شده از دو دیتاست قبلی بدین شکل شد :

1. Accuracy : 23.58 %
2. Precision : 25.20 %
3. Recall : 23.57 %
4. F1 Score : 23.26 %
5. Fitting Time : 41.46 Mins
6. Confusion Matrix :

[[367 7 80 8 69 147 135 148 14 5]  
 [ 0 420 28 650 0 0 32 3 2 0]  
 [ 24 33 549 29 126 25 98 133 10 5]  
 [ 15 114 95 109 107 94 373 16 49 38]  
 [ 6 330 107 68 27 31 110 36 75 192]  
 [ 36 98 86 52 14 202 326 30 31 17]  
 [ 13 10 459 28 1 5 146 292 1 3]  
 [ 0 270 11 50 1 4 12 95 268 317]  
 [ 26 120 39 18 5 245 317 20 168 16]  
 [ 1 308 12 35 2 24 7 13 332 275]]

**پی نوشت : در نهایت باز هم به دلیل دقت پایین تمرین عذرخواهی می کنم و باز هم ذکر می کنم به دلیل انجام تمرین در بازه امتحانات وقت کافی برای بهینه کردن آن را نداشتم.**