گزارش تمرین ۱

نام و نام خانوادگی: غزل رفیعی

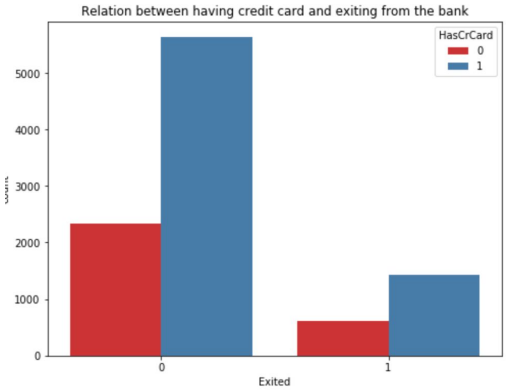
شماره دانشجویی: ۹۷۲۲۲۰۴۴

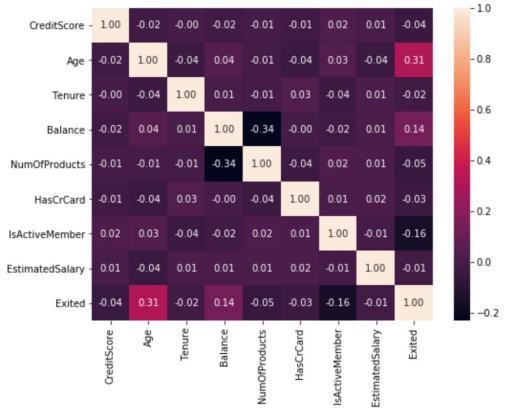
هدف، پیش‌بینی خروج شخص از بانک با استفاده از اطلاعات اوست که شامل موارد زیر می‌شود:

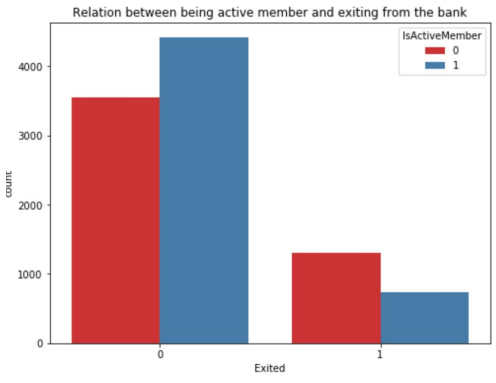
`CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited`

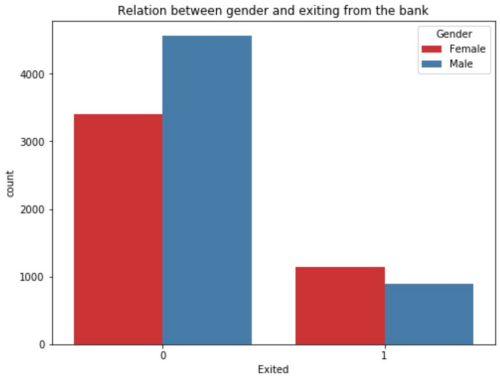
همان‌طور که در Python Notebook ضمیمه شده مشخص است، ابتدا بررسی شد آیا داده‌ی خالی در جدول وجود دارد یا خیر. و چون همه‌ی ردیف‌ها همه‌ی مقادیر را دارا بودند، از مرحله‌ی تخمین داده‌های null عبور کردیم.

برای بخش Visualization، ابتدا یک نقشه‌ی گرمایی(heat map) کشیده شد تا به طور دقیق هم‌بستگی بین هر دو متغیر مشخص شود و سپس چند نمودار بر حسب ستون `Exited` کشیده شد تا ببینیم متغیرهای متفاوت، چه ارتباطی با خارج شدن یا نشدن فردی از بانک دارند.









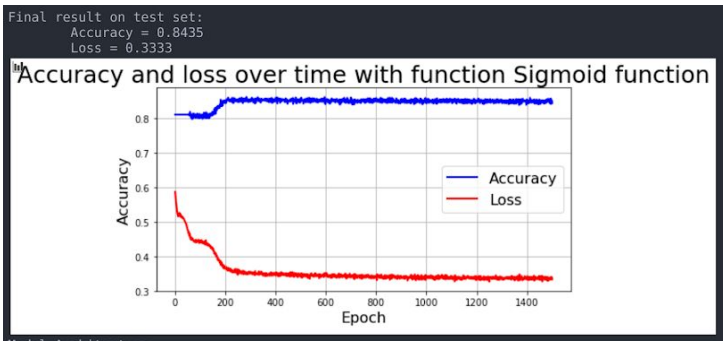
سپس ۱۰,۰۰۰ داده با نسبت ۰.۲ به داده های تست و یادگیری تقسیم شدند و در داده‌های یادگیری، ستون `Exited` حذف شد تا پیش‌بینی شوند. همچنین ستون‌هایی که حاوی مقادیر کیفی بودند،یعنی `Geography , Gender` با استفاده از روش زیر به بردار های کمّی تبدیل شدند.

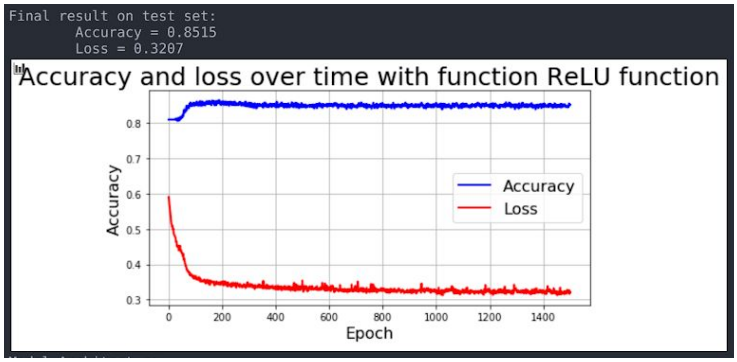
هر داده‌ی کیفی به برداری n بعدی تبدیل شد که n تعداد مقادیر ممکن بود و بعدی که داده‌ی کیفی برابر آن بود، مقدار ۱ و بقیه‌ی ابعاد، مقدار ۰ گرفتند.

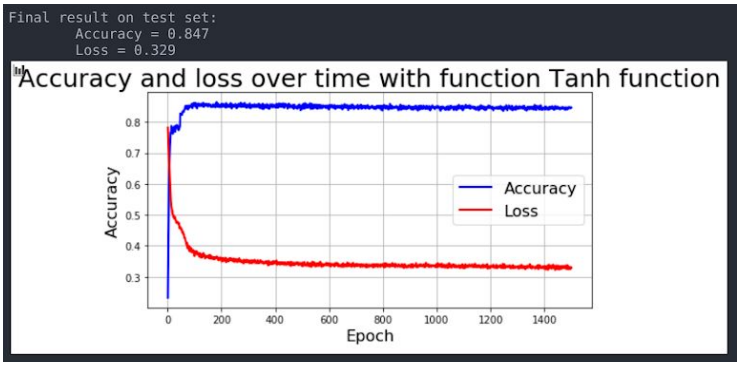
سپس همه‌ی داده‌ها با استفاده از کتابخانه‌ی Scikit Learn پایتون استاندارد (یا Normalize) شدند تا تاثیر مقادیر خاص هر پارامتر، نادیده گرفته شوند. یعنی به طور مثال داده‌ای که در بازه‌ی ۰ و ۱۰۰۰ قرار دارد، نباید تاثیر بیشتری در پیش‌بینی داشته باشد تا داده‌ای که در بازه‌ی -۱ و ۱ قرار دارد و به این منظور، همه‌ی داده‌ها با استفاده از تبدیل خطی، در بازه‌ی -۱ و ۱ قرار گرفتند.

برای شروع مرحله‌ی یادگیری شبکه‌ی عصبی، سه مدل ساده با ۵ لایه ساخته شد که در تمامی لایه‌های هر شبکه، به ترتیب از توابع ReLU و Tanh و sigmoid استفاده شد. اندازه‌ی لایه‌ها، به ترتیب ۱۱و۱۵و۱۲و۸و۴و۱ انتخاب شدند که ۱۱ و ۱ ، تعداد پارامترهای ورودی و خروجی هستند. برای مشاهده‌ی نتایج در تمامی مراحل، تابع `Print_Model_Result` ساخته شد که در آن برای محاسبه‌ی Loss، از تابع Binary Cross Entropy و Backpropagation و Adam برای بهینه‌سازی استفاده می‌شود. سپس در تعداد مشخصی epoch و با فاصله‌های مساوی، نتیجه‌ی تست و مقدار درستی و Loss داده‌های پیش‌بینی شده نمایش داده می‌شود و بر حسب آن‌ها، یک نمودار کشیده می‌شود.

نمودارها:





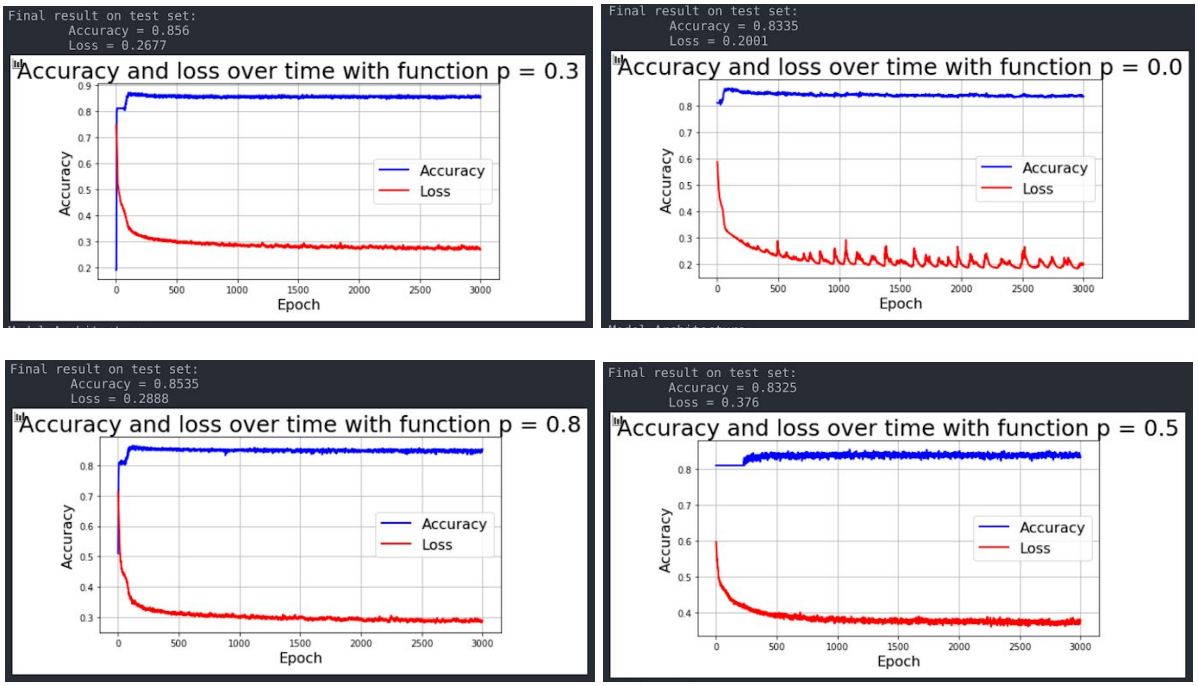


در این مرحله، بهترین نتایج، مربوط به شبکه با توابع ReLU و tanh است که از آنها در تست های بعدی استفاده می‌شود.

در قسمت بعد، لایه‌های Dropout و الگوریتم L2 Regularization به شبکه اضافه می‌شود تا از Overfitting جلوگیری شود.

لایه‌های Dropout یکی در میان در میان لایه‌های دیگر قرار میگیرند و احتمال‌های ۰.۰ و ۰.۳و ۰.۵و ۰.۸ برای صفر شدن یک element در نظر گرفته می‌شود تا مقایسه شود کدام یک بهتر عمل می‌کند. که با توجه به مقادیر به دست آمده، ۰.۰ و ۰.۳ با کمتر از ۲۰ epoch بهترین نتیجه را دارد.

نمودارها:

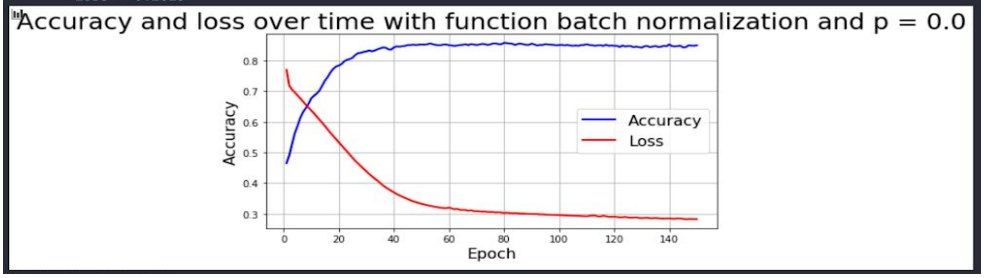


در بخش بعدی، لایه‌های Batch Normalization در میان برخی لایه‌ها قرار داده می شود. هدف این کار، استاندارد کردن دوباره‌ی داده‌ها است چون ممکن است در حین آموزش، از بازه‌ی -۱ و ۱ خارج شوند.

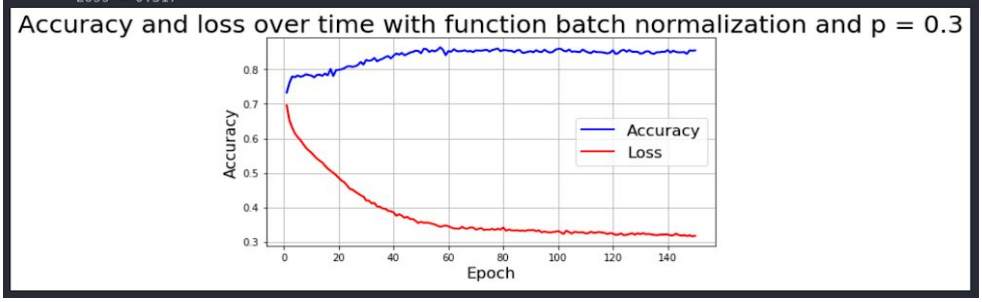
در این مرحله نیز مقادیر متفاوتی برای احتمال Dropout و epochs انتخاب شد و نتیجه به طور قابل توجهی تغییر نکردو حدود درستی(Accuracy)، همواره حدود ۰.۸۵ باقی ماند.

نمودارها:

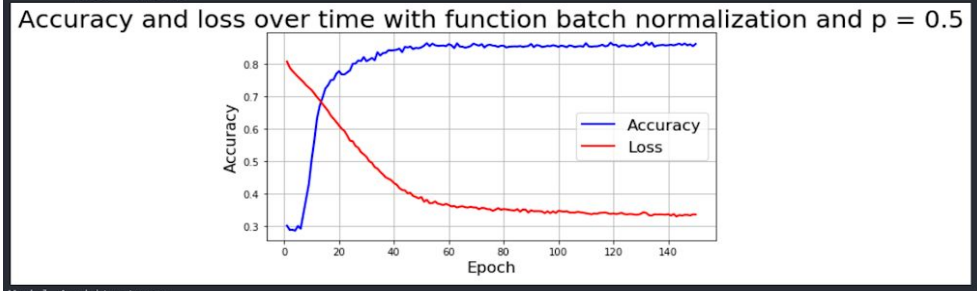
```
Final result on test set:
        Accuracy = 0.8495
        Loss = 0.2829
```



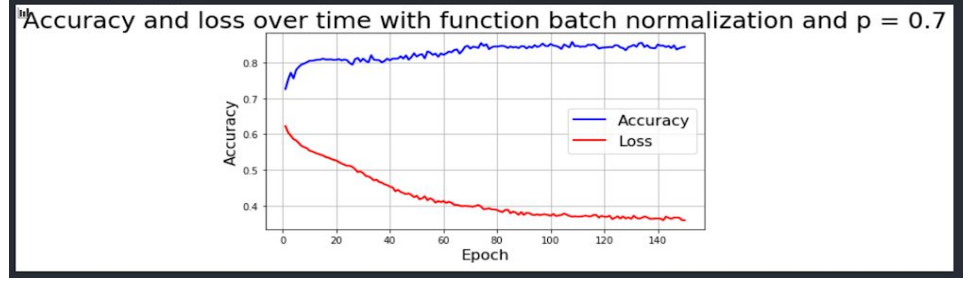

```
Final result on test set:
        Accuracy = 0.8555
        Loss = 0.317
```




```
Final result on test set:
        Accuracy = 0.8605
        Loss = 0.3348
```




```
Final result on test set:
        Accuracy = 0.844
        Loss = 0.3592
```



در نهایت نیز، بهترین مدل‌های به دست آمده به ترتیب روی سری داده‌های Validation تست شدند و نتایج زیر مربوط به درصد درستی پیش‌بینی آن‌هاست.

(تفاوت داده‌های Validation و Test در این است که داده های Validation در آموزش به کار برده نمی‌شوند بنابراین، شهود بهتری از تست کردن داده‌های برای داده‌های جدیدی که در آینده قرار است به مدل داده شوند، به ما می‌دهند.)

```
Final accuracy with model_relu on test-set: 0.8525
Final accuracy with model_tanh on test-set: 0.8615
Final accuracy with model_drop1 on test-set: 0.852
Final accuracy with model_batch1 on test-set: 0.845
Final accuracy with model_batch2 on test-set: 0.857
```

همان‌طور که در نتایج مشخص است، مدل اولیه که با استفاده از دو تابع ReLU و Tanh ساخته شد، بهترین مدل‌های ممکن بودند و اضافه کردن Dropout و Batch Normalization تاثیر قابل توجهی روی Accuracy نداشتند.