

بسم الله الرحمن الرحيم

دکتر جمال الدین گلستانی

درس یادگیری ماشین

پاسخ گزارش تمرین عملی سری 3

امیرحسین رستمی 96101635

دانشگاه صنعتی شریف

پرسش های بخش A:

1.3.2 A2. Perceptron Algorithm

In this part, you should implement the perceptron algorithm from the scratch. First, add one dimension with constant 1 to each data point (i.e. $x_0 = 1$). What is the purpose of this?

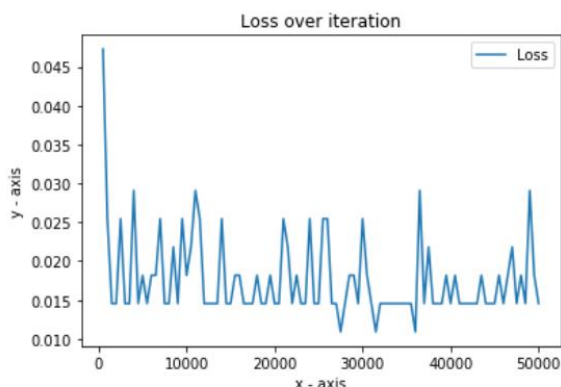
اضافه کردن این 1 باعث می شود که رابطه $w'x'$ به $w'x + b$ تبدیل می شود که یک رابطه خطی و محاسبات و معادلات را ساده تر می کند.

A6. Conclusion

discuss and compare the resulting weights and errors ($\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{prediction_i \neq y_i}$) of the above methods in your report.

ابتدا نمودارهای هر بخش را (که در فایل ژوپیتر) هم خروجی آن ها موجود است را بیان می کنم و سپس به مقایسه آن ها می پردازم.

1- نمودار خطا بر حسب iteration در perceptron (بدون اضافه کردن فیچر x_4^3)

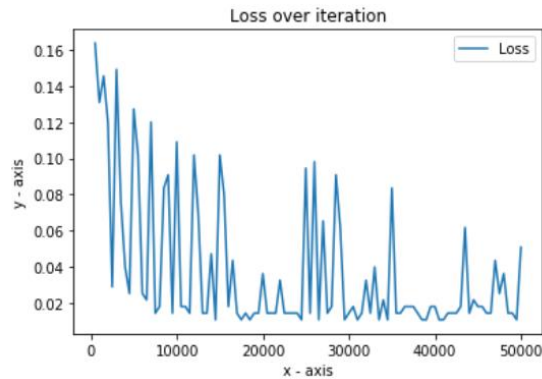


مقدار خطای برابر:

$$TrueRisk_{validationSet} = 0.014545$$

وجود نوسان نسبی در نمودار خطا به این معناست که مساله ما کاملاً از جنس linear separable نبوده است و به میزانی خصلت غیرخطی دارد، به اصطلاح مدل ما تا هوای یک دسته را نگه می دارد، دسته دیگر از تشخیص اش خارج می شود و دوباره تا می آید هوای دسته دیگر را داشته باشد دسته فعلی از تشخیصش خارج می شود و به همین دلیل دچار نوسان می شود. البته خطای ما در حد 1.5 درصد است و می توان از مدل به خوبی برای تشخیص استفاده کرد.

2- نمودار خطا بر حسب iteration در perceptron (با اضافه کردن فیچر x_4^3)



مقدار خطا برابر:

$$TrueRisk_{validationSet} = 0.050909$$

همانطور که مشاهده می کنید با اضافه کردن فیچر x_4^3 خطا در حدود 3.5 برابر شده است و این یعنی اینکه اضافه کردن این فیچر کمکی به بهتر شدن مدل ما نکرده است.

همانطور که می دانید از لحاظ علامتی x_4^3 با x_4 یکسان است و از آنجا که مقادیر x_4 اکثراً کوچک اند، اضافه کردن این فیچر آنقدر اطلاعات مازاد برای مدل به ارمغان نمی آورد و منطقی است که به بهبود مدل کمک نکرده است، علاوه بر آن، مشاهده کردید که در حالت بدون فیچر خطای ما در حد 1 درصد بود و خطای معقولی است و وقتی ما فیچر x_4^3 را که یک فیچر غیرخطی از فیچرهای قبلیست در دیتا اضافه می کنیم به نوعی به مدلی که در حد خوبی خطی بود، استیت غیرخطی اضافه کرده ایم و این باعث شده است مدل ما از فرض linear separable بودن بیشتر فاصله گرفته است، لذا خطای تشخیص مدل ما که بر مبنای perceptron است افزایش پیدا کرده است.

3- SVM (حالتی که فیچر اضافه، وارد نکرده ایم).

Emperical Loss:

0.010018214936247723

True Loss:

0.014598540145985401

Weight:

[[-1.74782541 -1.00507855 -1.22904841 -0.14600023]]

همانطور که مشاهده می کنید، خطای واقعی ما تا حد خوبی برابر با خطای واقعی حالت مشابه با perceptron است و این یعنی اینکه توانمندی این دو مدل در تشخیص این دیتاست تا حد خوبی برابر با هم بوده است.

SVM - 4 (در حالتی که فیچر جدید اضافه کرده ایم).

Emperical Loss:

0.01092896174863388

True Loss:

0.014598540145985401

Weight:

$[[-2.45950847e+00 \ -1.41875226e+00 \ -1.72020474e+00 \ -1.92110426e-01$
 $-6.69808618e-04]]$

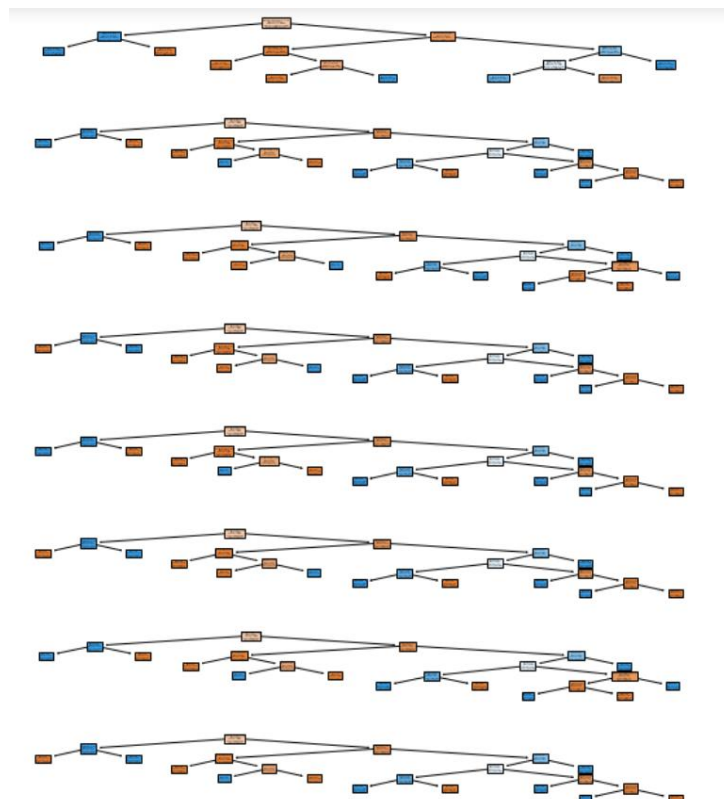
همانطور که مشاهده می کنید، در این حالت خطای ما نسبت به حالت قبلی افزایش پیدا کرده است و دلایلی مشابه با آنچه در قسمت 2 آورده ایم را هم چنان می توان برای علت این افزایش خطا، ذکر کرد.

پرسش های بخش B:

Set the maximum depth of the tree to $\{4, 6, 8, 10, 12, 14, 16, 18, 20\}$. For each maximum depth, train a classifier on the training data and report the resulting loss on the validation set. Plot the loss against the maximum depth. What is the best maximum depth? Finally, for the best maximum depth, report the loss on the test set.

برای مشخص شدن اینکه کدام ماکزیمم عمق بهتر است، پس از امتحان کردن هر کدام درختی آن را رسم می کنیم تا ببینیم کدام ماکزیمم عمق برای مدل مناسب است، در قسمت زیر نمودار درختان را به ترتیب افزایش ماکزیمم عمق مشاهده می کنید:

"نمودار درختان به ترتیب از بالاترین با ماکزیمم عمق 4 تا پایین ترین با ماکزیمم عمق 20"



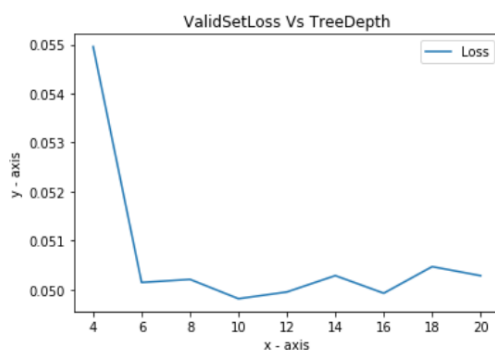
همانطور که مشاهده می کنید حتی با افزایش ماکزیمم عمق، در نهایت عمق درختمان 6 شده است و لذا داریم که بهترین ماکزیمم عمق برای این دیتاست 6 است.

همانطور که مستحضر هستید چون تی ای محترم فرمودند که متغیر random_state مدل decision tree را تغییر ندهیم و این باعث می شود که در هر مرحله اجرا داریم که یک Randomness در مدل ما وجود دارد و این باعث می شود که مقایسه خطای عمق های مختلف سخت شود، برای حل این مشکل همانطور که تی ای محترم نیز اشاره کردند، به ازای تعداد مراحل خوبی تست گرفتیم و میانگین خطای محاسبه شده را به عنوان خطای در نظر میگیریم و آن را تحلیل می کنیم. همانطور که ملاحظه کردید بهترین "ماکزیمم عمق" برابر 6 است و اصلا مدل های با عمق بیشتر، از 6 لایه بیشتر نمی گردند و لذا خیلی انتظار معقولیست که خطای برای ماکزیمم عمق های 6 به بعد به صورت تقریبی با هم برابر شود و همه این اعماق، خطای میانگینشان نسبت به مدل با ماکزیمم عمق 4، کمتر خواهد بود.

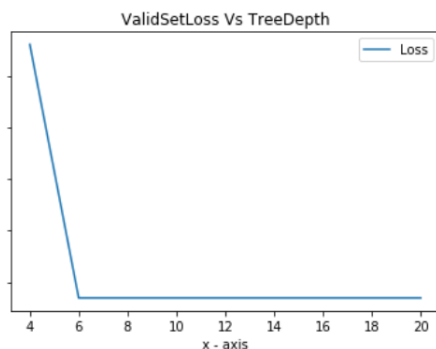
علت:

واضح است که وقتی اجازه می دهیم که ماکزیمم عمق درخت تصمیم گیری ما بیشتر می شود، جزییات بیشتری از ساختار دیتا را می توانیم درون درخت ذخیره کنیم و لذا منطقیست که خطای ما کاهش پیدا کند.

نمودار خطای درختان تصمیم گیری بر حسب متغیر ماکزیمم عمق:



توجه کنید که اگر تعداد iteration ها رو جهت محاسبه خطای میانگین به اندازه کافی بزرگ بگذارید نمودار به حالت زیر تبدیل می گردد:



همانطور که مشاهده می کنید مطالبی که پیشتر برای این نمودار پیش بینی کردیم تماما صادق و برقرار است.