

تشخیص بهمن های نوروئی بحرانی با استفاده از روش های یادگیری ماشینی

فاطمه سلیمانی مقدم، علی ابوالحسن زاده ماهانی، علی قبله، روژین محمدی کیان

استاد درس: دکتر صادق رئیسی

استاد ناظر: دکتر سامان مقیمی عراقی

۳۰ دی ۱۴۰۰

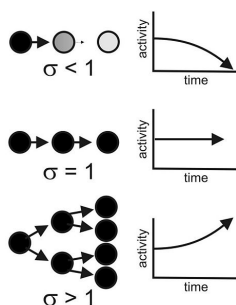
فهرست مطالب

۳	۱ طرح مسئله
۴	۲ داده ها و پیش پردازش
۵	۳ استفاده از روش های سنتی یادگیری ماشینی
۱۴	۴ استفاده از شبکه های عصبی
۱۵	۵ نتیجه گیری

۱ طرح مسئله

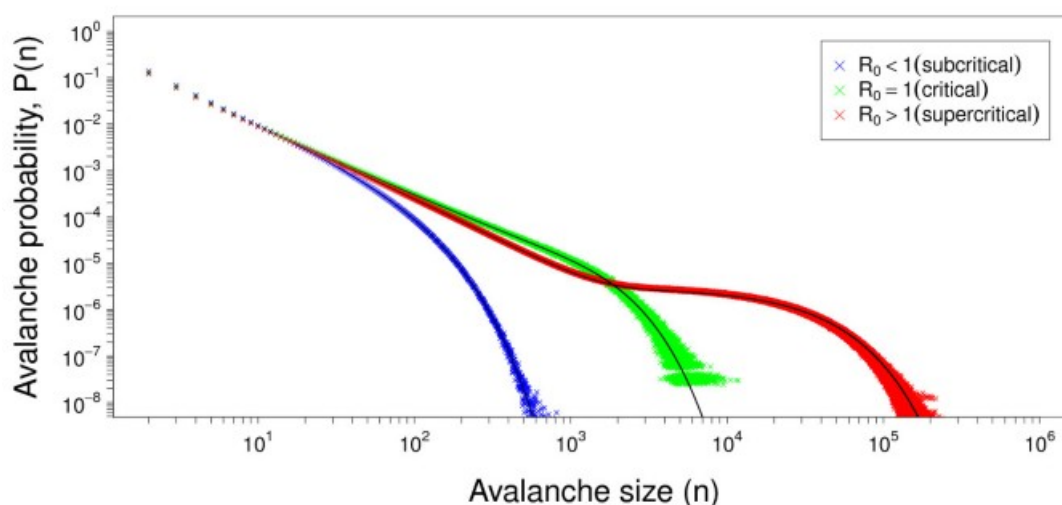
بهمن های نورونی، یک سلسله فعالیت های نورونی است که از یک نقطه شروع شده، ادامه پیدا می کند و تمام می شود، مانند یک شعله که از یک کبریت روشن شده و کبریت های اطراف خود را روشن می کند. برای توصیف این پدیده، پارامترهایی داریم مانند اندازه ی بهمن (مجموع نورون های درگیر در هر بهمن)، زمان بهمن و پارامترهای خارجی دیگر مانند γ و g . این پارامترهای خارجی را از زمان و اندازه ی نورون جدا می کنیم زیرا چیزی که در عمل برایمان قابل اندازه گیری است، زمان بهمن و اندازه ی بهمن می باشد.

نشان داده شده است که تابع توزیع احتمال برای اندازه ی بهمن ها در یک حالت خاص، با اندازه و زمانی بهمن رابطه ی نمایی دارد [۱] که به آن حالت بحرانی می گوئیم. در حالت بحرانی، هر نورون فعال در زمان t ، فعالیت خود را فقط به یک نورون در زمان $t+1$ انتقال می دهد و تعداد نورون های فعال در هر لحظه ثابت خواهد بود. حالت های دیگر هم داریم، حالت فرو بحرانی و فرابحرانی که در آن ها شدت فعالیت کل نورون ها در زمان به ترتیب کاهش و افزایش می یابد.



شکل ۱.۱: نحوه ی انتقال فعالیت در بهمن های فروبحرانی، بحرانی و فرا بحرانی

اما باید توجه داشته باشیم که مطلب بالا برای یک جمعیت نورونی با تعداد نورون بی شمار صادق است. برای جمعیت های متناهی (مانند مغز)، بالاخره وقتی قله ی موج تیزه زدن به آخرین نورون ها رسید، فعالیت پس از مدتی می میرد، چه حالت بحرانی داشته باشد چه فرابحرانی. بهمن فرو بحرانی هم که علی الاصول چه جمعیت متناهی داشته باشیم چه نا متناهی، پس از مدتی خاموش می شود. اگر نمودار هر سه حالت را رسم کنیم، می بینیم که احتمال رخ دادن بهمن با اندازه های خیلی بزرگ صفر است؛ همین طور احتمال رخ دادن بهمن با زمان های خیلی طولانی. سیستم های عصبی که پارامترهایشان روی حالت بحرانی تنظیم شده است در زمینه های محاسبات، انتقال و ذخیره ی اطلاعات بهترین عملکرد را دارند. بنابراین دانشمندان علوم اعصاب تمایل دارند حالت های بحرانی را تشخیص دهند. اما به دلیل گفته شده تشخیص این حالت برای جمعیت های نورونی متناهی راحت نیست.



شکل ۲.۱: توزیع احتمال بر حسب اندازه ی بهمن برای سه حالت فروبحرانی، بحرانی و فرابحرانی [۲]

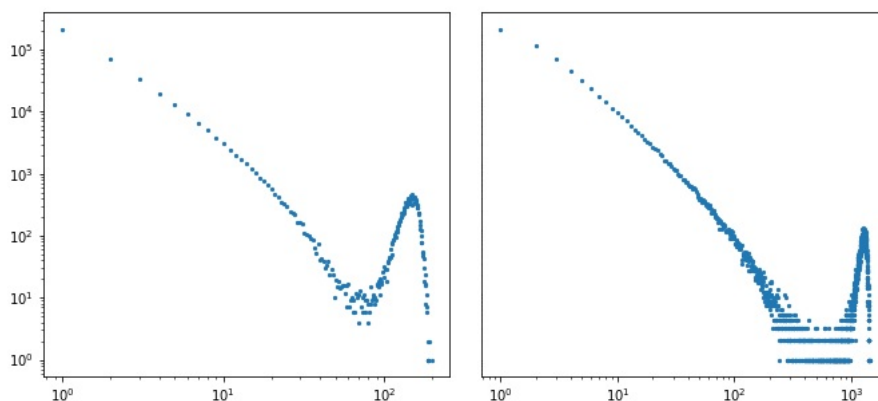
اگر توزیع احتمال اندازه و زمان بهمن ها را به صورت تمام لگاریتمی رسم کنیم، می بینیم که حالت بحرانی تقریباً با چشم قابل تشخیص است. اما این کار برای هزاران داده ای که دانشمندان علوم اعصاب می خوانند با آن ها کار کنند امری طاقت فرسات. بنابراین استفاده از روش های یادگیری ماشینی کمک مفیدی در این زمینه می تواند باشد.

۲ داده ها و پیش پردازش

داده های مورد استفاده در این پروژه از شبیه سازی رایانه ای گرفته شده و بیش از ۲۰۰ حالت مختلف از پارامترهای خارجی را در بر می گیرد، هرچند این پارامترها را در مدل های مان استفاده نمی کنیم، اما از این جهت این تعداد تنوع خوب است که مدل ما بتواند سوای پارامترهای خارجی جمعیت نورو، بحرانی بودن یک بهمن را تشخیص دهد.

پیش پردازش داده ها

در شبیه سازی بهمن ها، محدودیت های محاسباتی وجود دارد. یکی از آن ها این است، محدودیت دفعات شبیه سازی است؛ که مثلاً اگر ۱۰۰۰ بار یک بهمن را شبیه سازی کنیم، شاید آن اندازه از بهمن که احتمال رخ دادنش کمتر از یک هزارم بوده است اصلاً رخ ندهد. بنابراین کامپیوتر آن مقدار آن را صفر خواهد گذاشت که طبیعتاً در نهایت برای آن اندازه ی بهمن، احتمال صفر در نظر گرفته می شود. مشکل بعدی، لگاریتمی کردن نمودار است؛ نمی توانم در تابع لگاریتم صفر بگذاریم، بنابراین مجبوریم یک مقدار کم به داده هایی که احتمال صفر دارند اضافه کنیم. در نهایت این دو موضوع باعث می شود که خیلی از داده هایی که احتمال رخدادشان به نسبت کم است (سمت راست نمودارهای شکل ۱.۲) احتمال یکسان با همسایه اشان داشته باشند و در نمودار سمت راست به صورت خط های افقی دیده شوند.



شکل ۱.۲: - نمودار یک بهمن فرابحرانی - سمت راست: نموداری قبل از پیش پردازش؛ سمت چپ: نمودار پس از پیش پردازش و سببندی

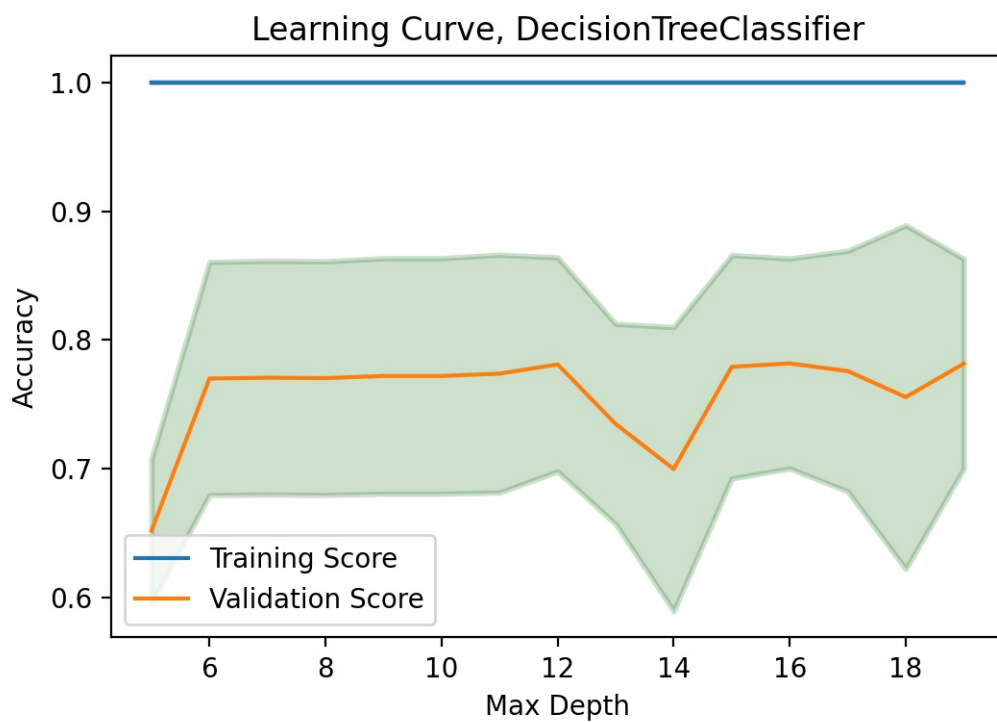
برای رفع این مشکل، کاری که کردیم، سبببندی داده ها بود؛ یعنی نقاط همسایه را در سبدهایی که روی محور افقی درست کرده ایم گذاشته و بسته به مقدار اعضای سبد، در مرکز سبد میانگین آن ها را می گذاریم (شکل ۱.۲ سمت راست). درست است که تراکم داده ها به این شکل کم می شود، اما از اول هم تراکم داده مسئله ای برای مان نبوده و حداقل به این صورت می توانیم ویژگی ها (features) داده هایمان را مانند شیب خط و استخراج کنیم.

استخراج ویژگی ها

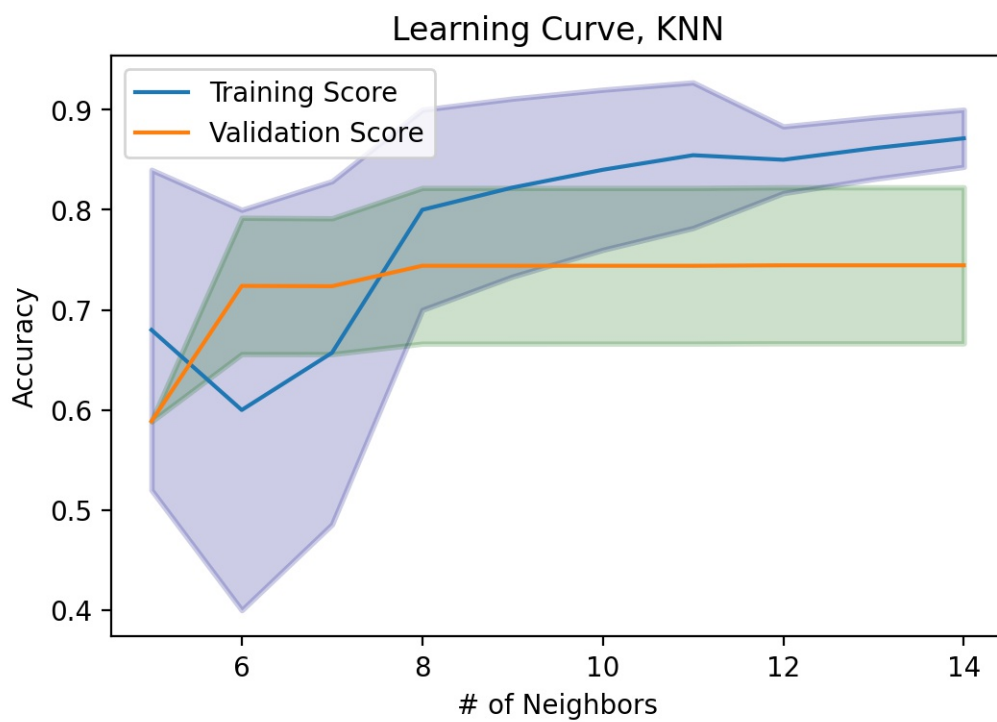
اگر به شکل ۲ توجه کنید، می بینیم که یک منحنی بهمن فروبحرانی، تقریباً در هیچ بازه ای به صورت خطی نیست. اما بهمن بحرانی تقریباً در بیش از ۸۰ درصد محور افقی به صورت خطی کاهش می یابد. پس شیب خط و مشتقات منحنی ها می توانند ویژگی های خوبی برای تمایز حالت فروبحرانی و بحرانی باشند. از طرفی، حالت فرابحرانی، مانند حالت بحرانی، در یک بازه ی قابل توجهی به صورت خطی نزول می کند، با این تفاوت که در منحنی آن اکسترم های موضعی می بینیم. پس با استفاده از کتابخانه ی `scipy.signal` و یافتن تعداد و همچنین مختصات اکسترم های منحنی ها، می توانیم دسته ی دیگری از ویژگی های را برای داده هایمان بسازیم.

۳ استفاده از روش های سنتی یادگیری ماشینی

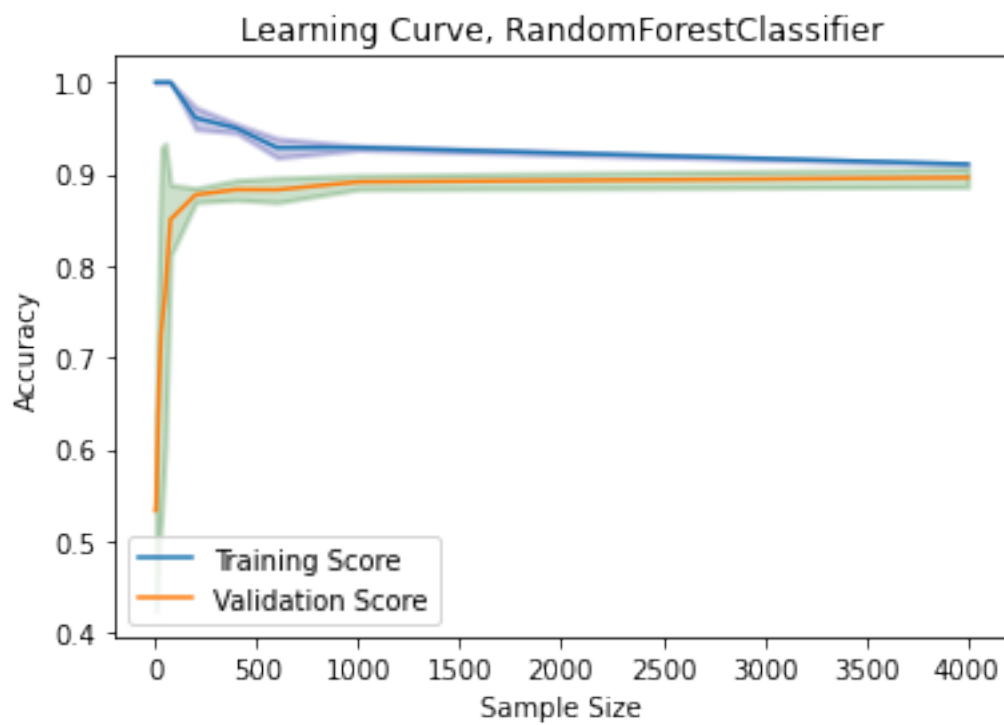
در این مرحله از ۵ optimizer مختلف استفاده کردیم، `SGD`، `KNN`، `SVM DT` و `RandomForest`. در تمام حالت های استفاده شده، طبق نمودارهایی که در ادامه آمده، دقت مدل تفاوت چندانی نداشتند، بنابراین برتری مدل ها تا حدی صرفاً وابسته به زمان یادگیری و زمان تست است.



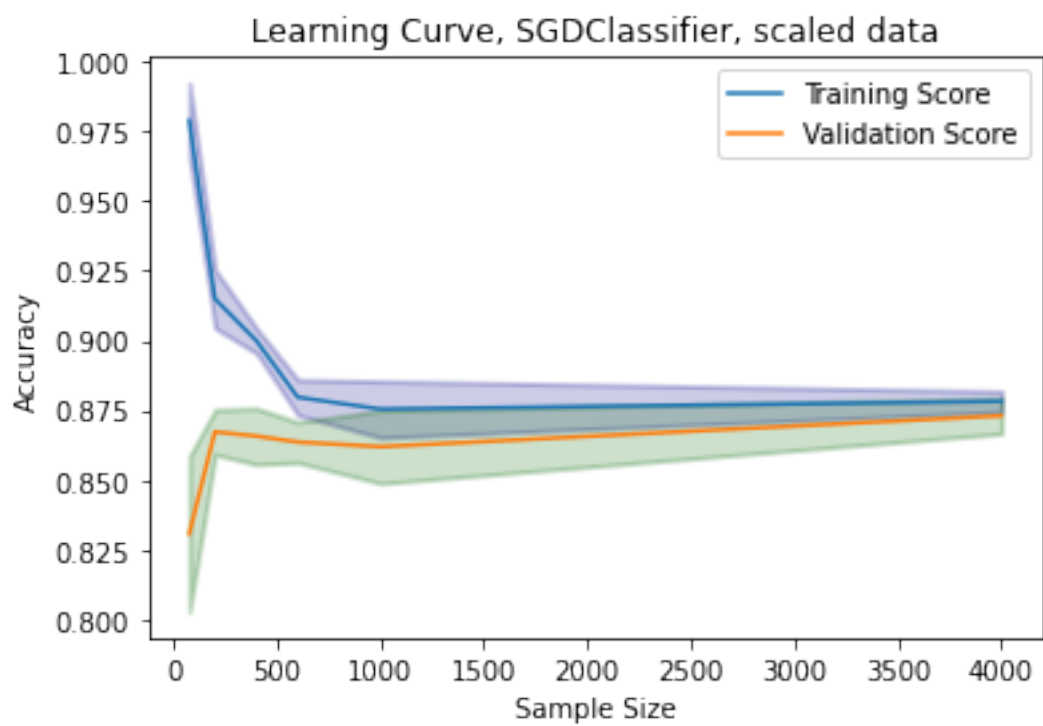
شکل ۱.۳: منحنی یادگیری برای درخت تصمیم گیری



شکل ۲.۳: منحنی یادگیری برای KNN

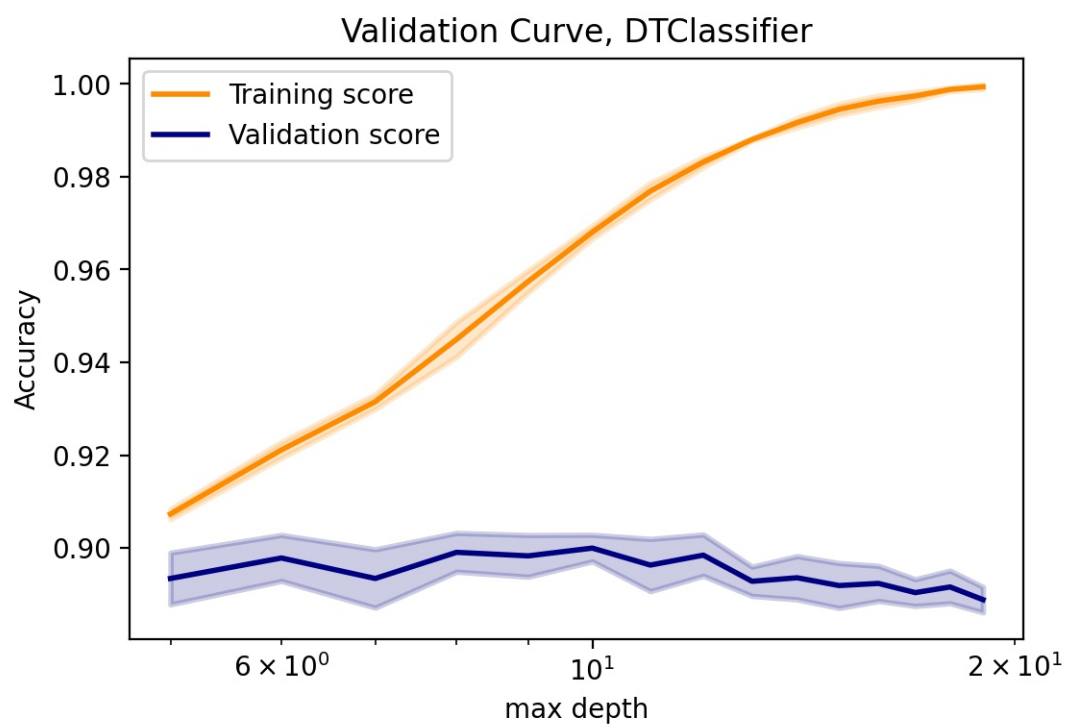


شکل ۳.۳: منحنی یادگیری برای جنگل تصادفی

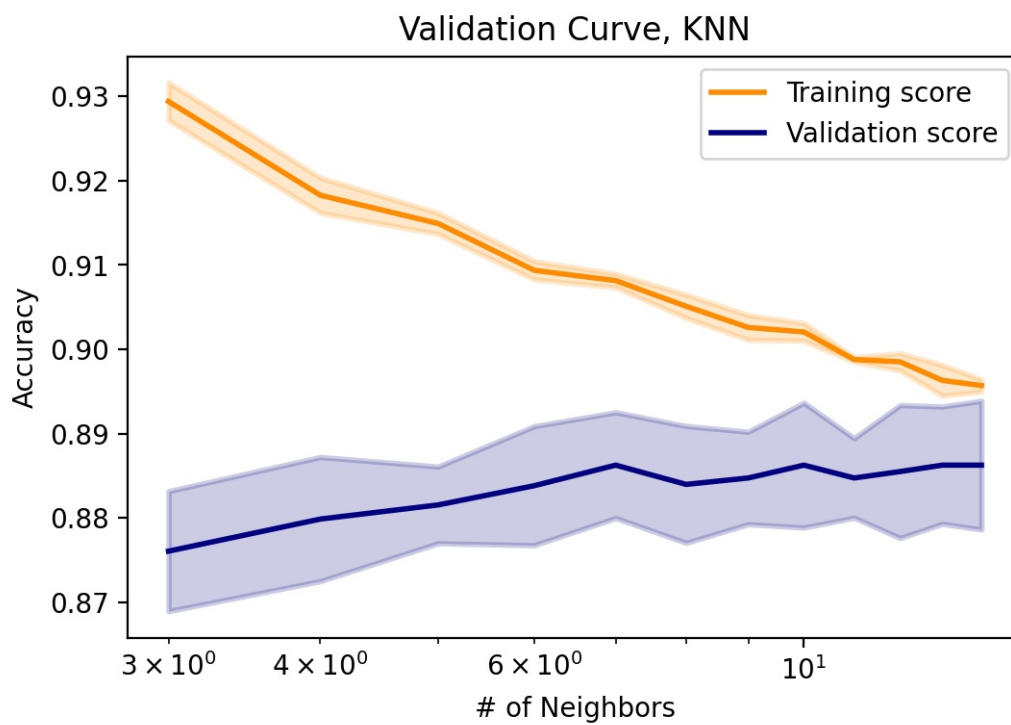


شکل ۴.۳: منحنی یادگیری برای SGD

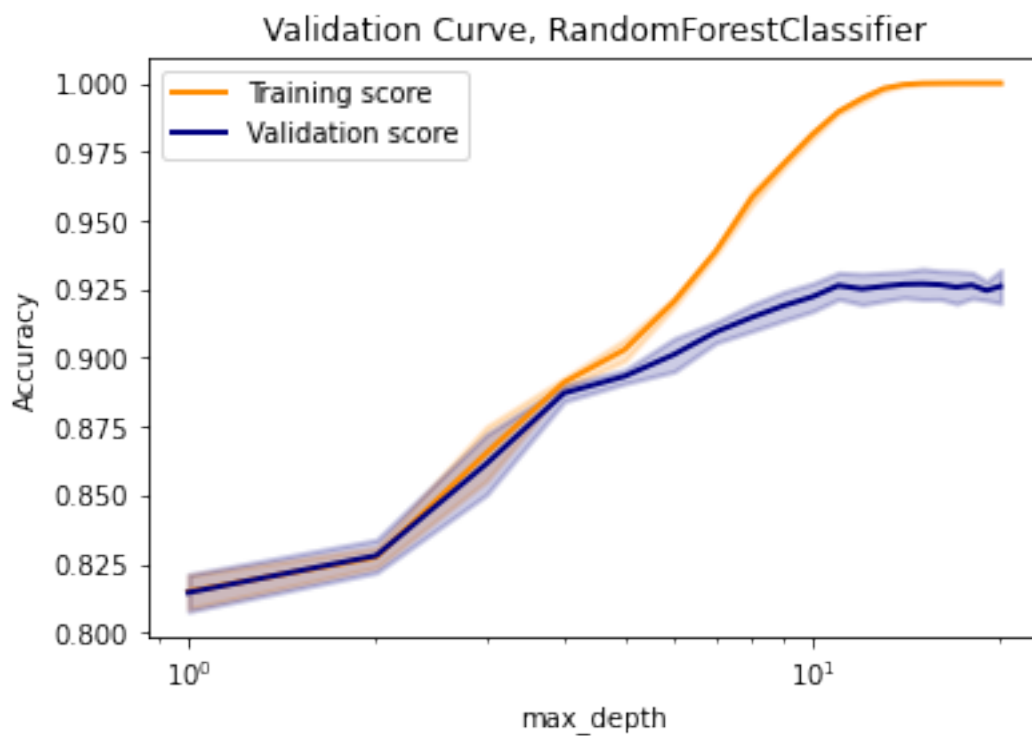
برای تمام مدل‌ها از داده‌های scale شده استفاده کردیم، زیرا برخی الگوریتم‌ها به شدت روی این مسئله حساس‌اند، برای مثال تفاوت curve learning برای الگوریتم sgk بین دو داده‌ی scale شده و scale نشده به شدت محسوس بود. متغیر این الگوریتم‌ها نیز همگی از طریق بررسی curve validation انتخاب شده‌اند. که به صورت زیر است.



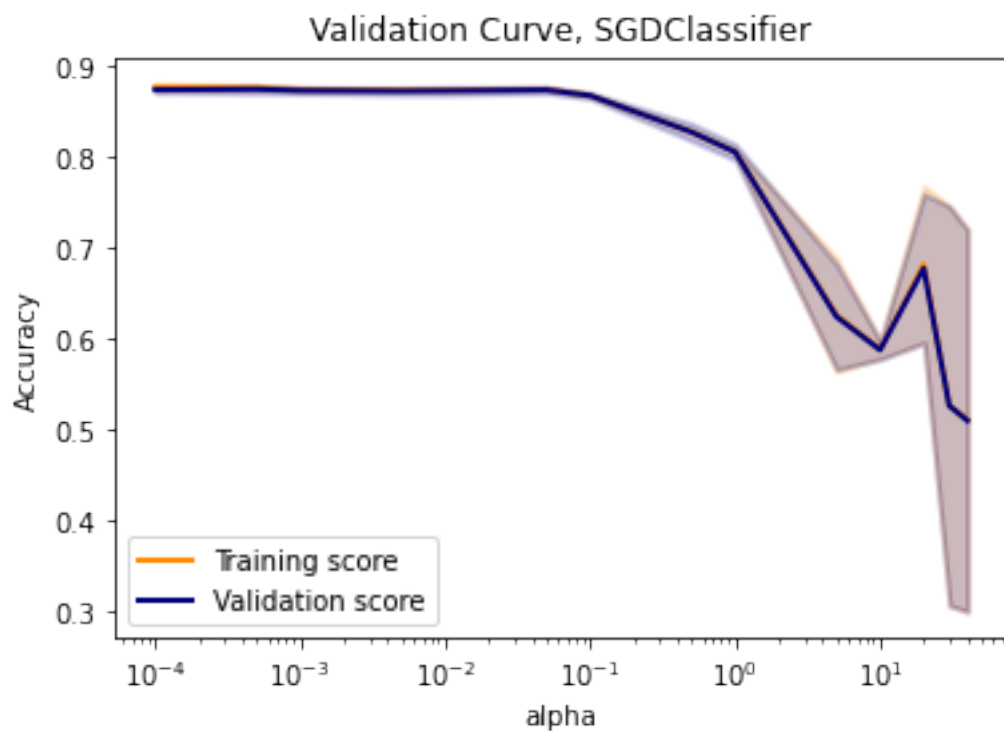
شکل ۵.۳: منحنی اعتبارسنجی برای درخت تصمیم‌گیری



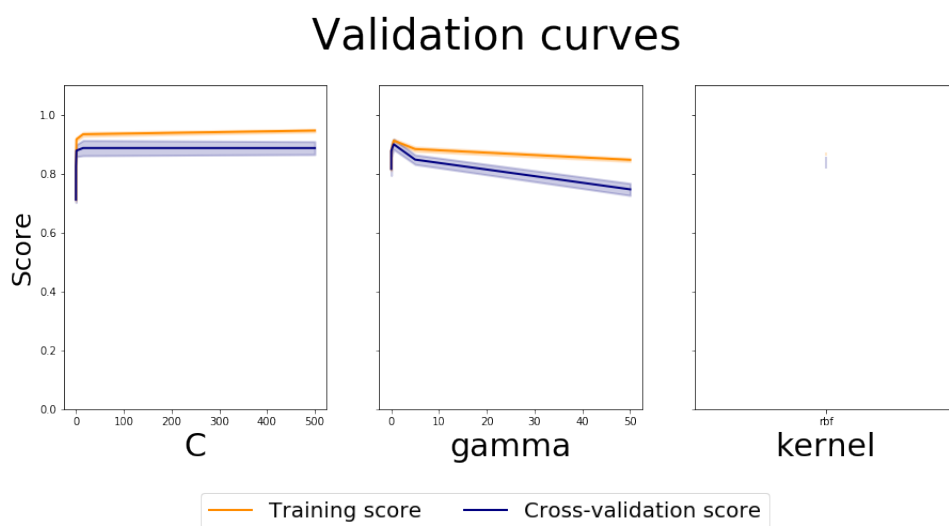
شکل ۶.۳: منحنی اعتبارسنجی برای KNN



شکل ۷.۳: منحنی اعتبارسنجی برای جنگل تصادفی



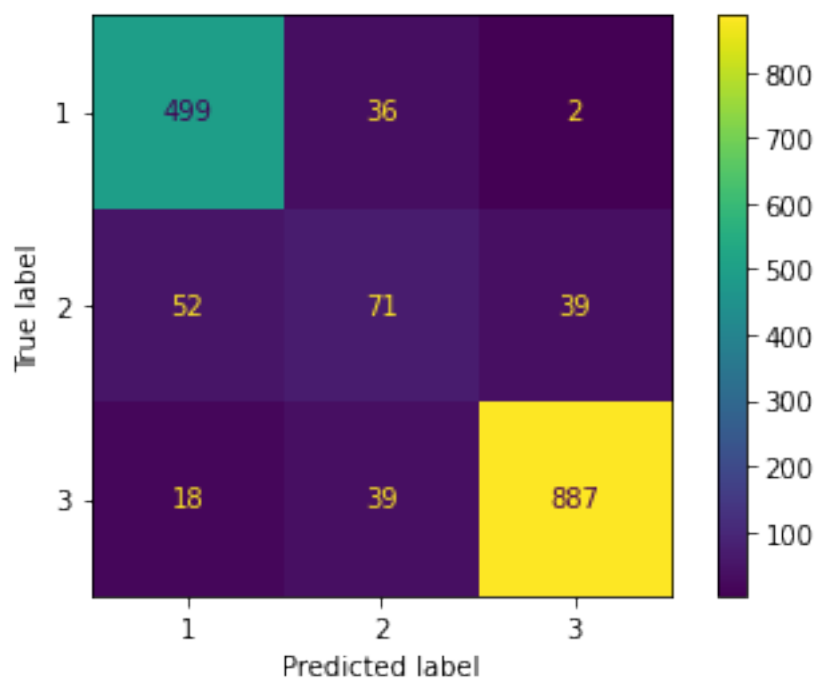
شکل ۸.۳: منحنی اعتبارسنجی برای SGD



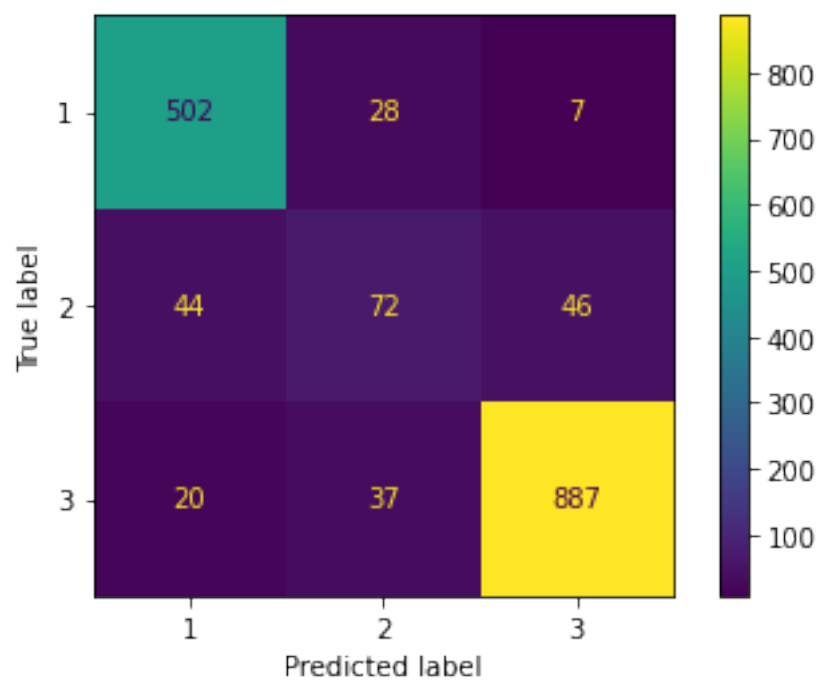
شکل ۹.۳: منحنی اعتبارسنجی برای SVM

برای مقایسه بهتر الگوریتم‌های استفاده شده، matrix confusion و جدول زمانی آن‌ها در ادامه آمده است.

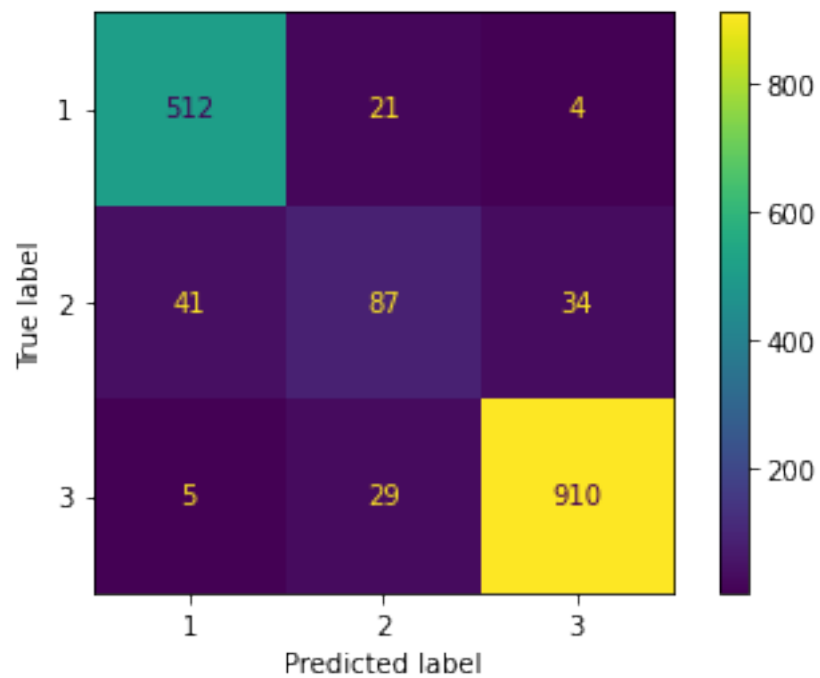
	SVM	SGD	RF	KNN	DT	Model
	۶۳۹	۴.۶۲	۱۴.۱	۸.۱۳	۵۵	Learning Time (ms)
	۱۸۰	۹۴.۱	۹.۷۰	۵.۸۶	۲	Test Time (ms)



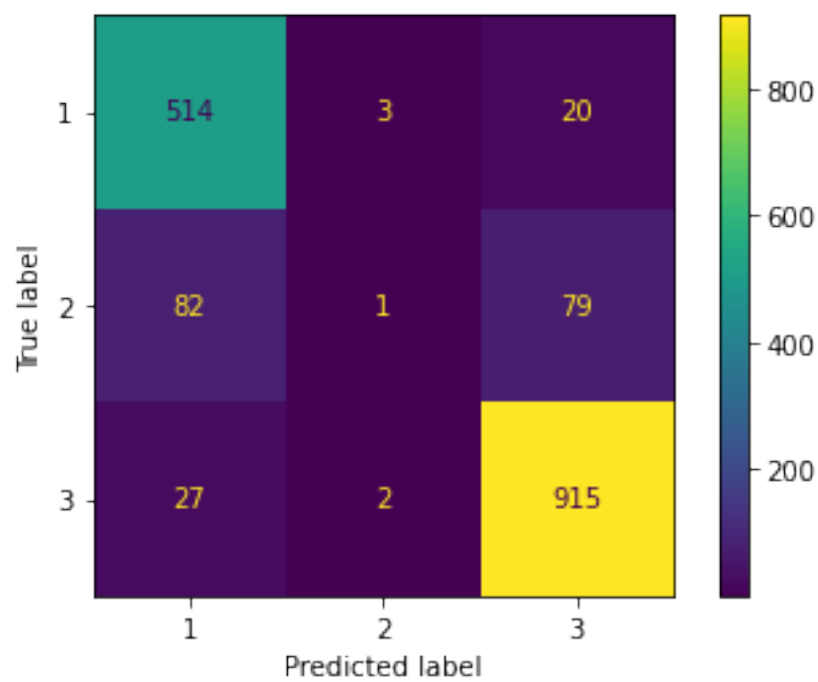
شکل ۱۰.۳: ماتریس درهم‌ریختگی برای درخت تصمیم‌گیری



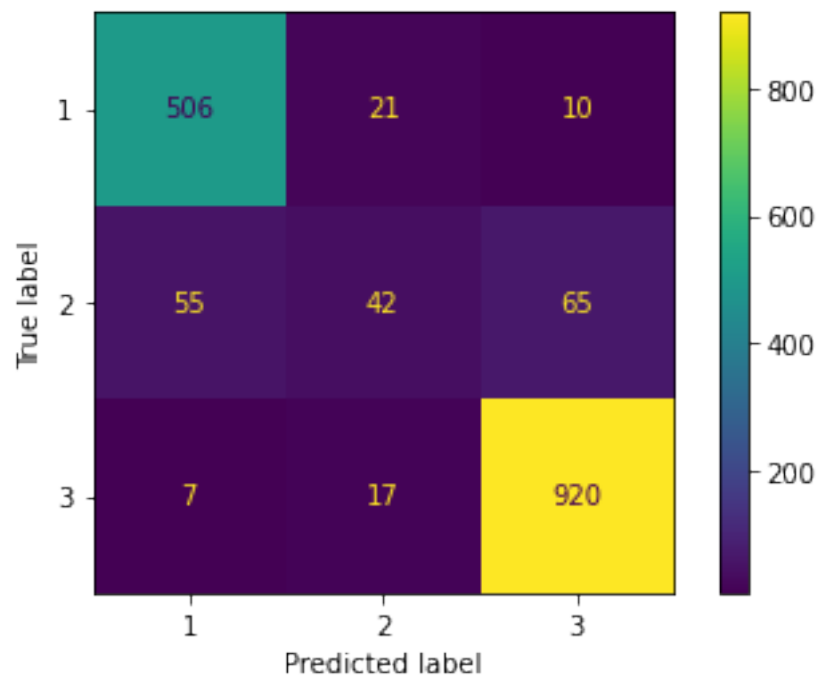
شکل ۱۱.۳: ماتریس درهم‌ریختگی برای KNN



شکل ۱۲.۳: ماتریس درهم‌ریختگی برای جنگل تصادفی



شکل ۱۳.۳: ماتریس درهم‌ریختگی برای SGD



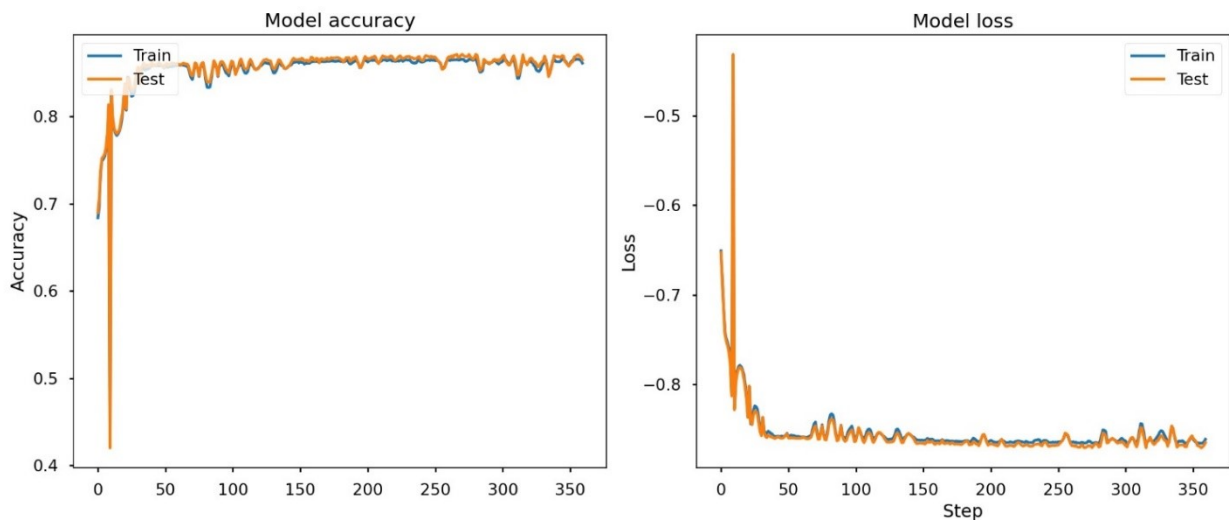
شکل ۱۴.۳: ماتریس درهم‌ریختگی برای SVM

۴ استفاده از شبکه‌های عصبی

در این قسمت، دو مسیر متفاوت را پیش گرفتیم، یکی به وسیله‌ی model sequential و دیگری با استفاده از pytorch از آنجایی که مسیر pytorch در آخر دقت (accuracy) بیشتری داشت، در ادامه توضیحات مربوط به این مدل را ارائه می‌کنیم. بعد از مرتب کردن داده‌ها و تبدیل آرایه‌های numpy به تنسور، یک شبکه عصبی با ۳ لایه (۲ لایه hidden و یک لایه output) تشکیل دادیم. از آنجاییکه داده‌ها دارای ۱۰ ویژگی بودند، تعداد گره‌های لایه اول ۱۰ و خروجی نهایی هم ۳ گره بود. چون کلاس‌بندی داده‌ها باینری نیست نیز از maxsoft استفاده کردیم. علاوه بر آن برچسب داده‌ها را هم به دلیل استفاده از

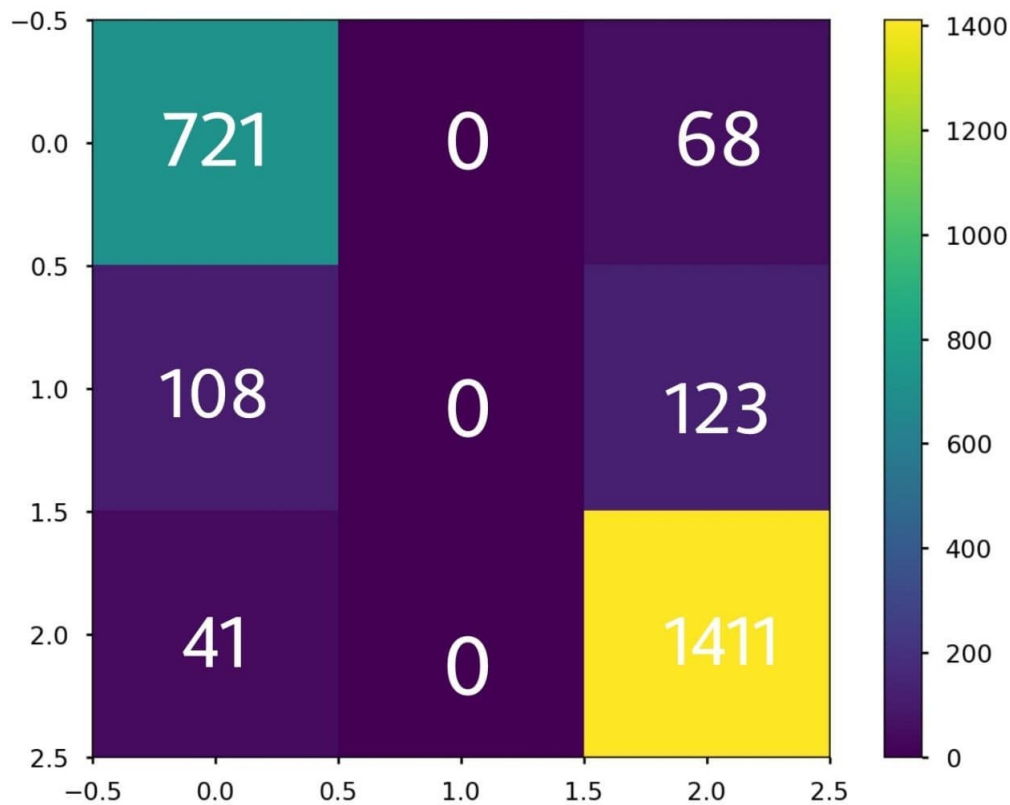
علاوه بر آن برچسب داده‌ها را هم به دلیل استفاده از max، argmax منهای یک کردیم تا با اندیس‌ها برابر باشد. در مدل از چند optimizer مختلف استفاده کردیم که نتایج مربوط به بهینه ساز ADAM مناسب‌تر بنظر می‌رسید. برای مثال برای ۵۰ گام و ۱۲ مینی‌بچ در مقایسه با SGD داریم:

Accuracy	Time Run	Optimizer
۸۵%	۵۵s	SGD
۹۰%	۵۰s	ADAM



شکل ۱.۴: نمودار دقت و هزینه‌ی مدل بر حسب تعداد داده

در ماتریس درهم‌ریختگی (confusion matrix) نیز می‌بینیم که برای کلاس بحرانی (۱)، هیچ پیش‌بینی درستی رخ نمی‌دهد. ممکن است این اتفاق به دلیل تعداد داده‌های کم این کلاس باشد. (به نسبت باقی کلاس‌های فروبحرانی و فرابحرانی برای این کلاس داده‌های خیلی کمی داشتیم هرچند که در طراحی مدل وزن این کلاس را بیشتر کردیم اما نتیجه‌بخش نبود).



شکل ۲.۴: ماتریس درهم‌ریختگی شبکه‌ی عصبی

۵ نتیجه‌گیری

با توجه به نتایجی که از هر کدام از مراحل گرفتیم و تعداد کم داده‌های کلاس بحرانی، بنظر می‌رسد که روش‌های سنتی به قدر کافی برای کلاس‌بندی داده‌های ما کافی هستند. دقت هر دو روش سنتی و شبکه عصبی نیز تفاوت چندانی نداشت، زمان هر دو روش تقریباً برابر بود اما در ماتریس درهم‌ریختگی مربوط به شبکه عصبی، نتایج مربوط به کلاس بحرانی اصلاً خوب نبود. هرچند که این اتفاق از قبل هم قابل پیش‌بینی بود زیرا تعداد داده‌های هر کلاس متقارن نبود و کلاس بحرانی‌های داده‌های بسیار کمی داشت. با کارهایی مثل تغییر وزن این کلاس و... باز هم نتیجه‌ای مناسبی نگرفتیم. علاوه بر آن روش‌های سنتی روش‌های ساده‌تری بودند و در این مسئله‌ی خاص، به قدر کافی نتایج خوبی داشتند.

مراجع

- [1] Poil SS, Hardstone R, Mansvelder HD, Linkenkaer-Hansen K. *Critical-state dynamics of avalanches and oscillations jointly emerge from balanced excitation/inhibition in neuronal*

networks. J Neurosci. 2012;32(29):9817-9823. doi:10.1523/JNEUROSCI.5990-11.2012

- [2] Taylor, T.J., Hartley, C., Simon, P.L. et al. Identification of *Criticality in Neuronal Avalanches: I. A Theoretical Investigation of the Non-driven Case*. J. Math. Neurosc. 3, 5 (2013). [https://doi.org/10.1186/2190-8567-3-](https://doi.org/10.1186/2190-8567-3-5)