به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

تمرین سری اول

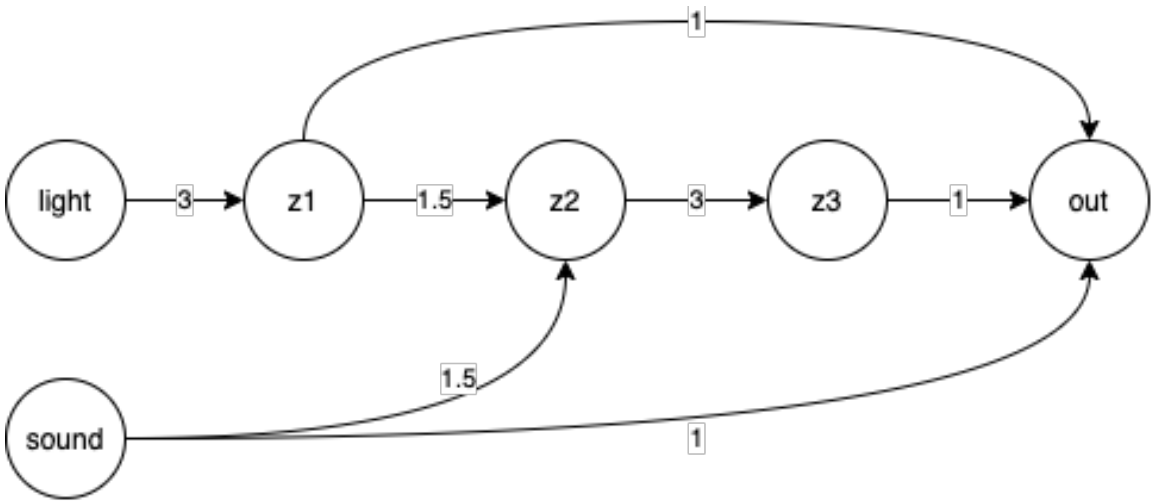
نام و نام خانوادگی	ژیوار صورتی حسن‌زاده
شماره دانشجویی	810196502
تاریخ ارسال گزارش	1400/01/10

فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را بهروز کنید.)

سوال ۱ – McCulloch-Pitts

الف) شبکه مد نظر را طراحی کنید.

ب) معادله منطقی این شبکه را بنویسید.



$$Thereshold = 3$$

$$\mathbf{out = sound(t\text{-}1)\ AND\ light(t\text{-}2)\ AND\ \{sound(t\text{-}3)\ AND\ light(t\text{-}4)\}}$$

شکل مربوط به شبکه تشخیص گفته شده به همراه معادله منطقی آن

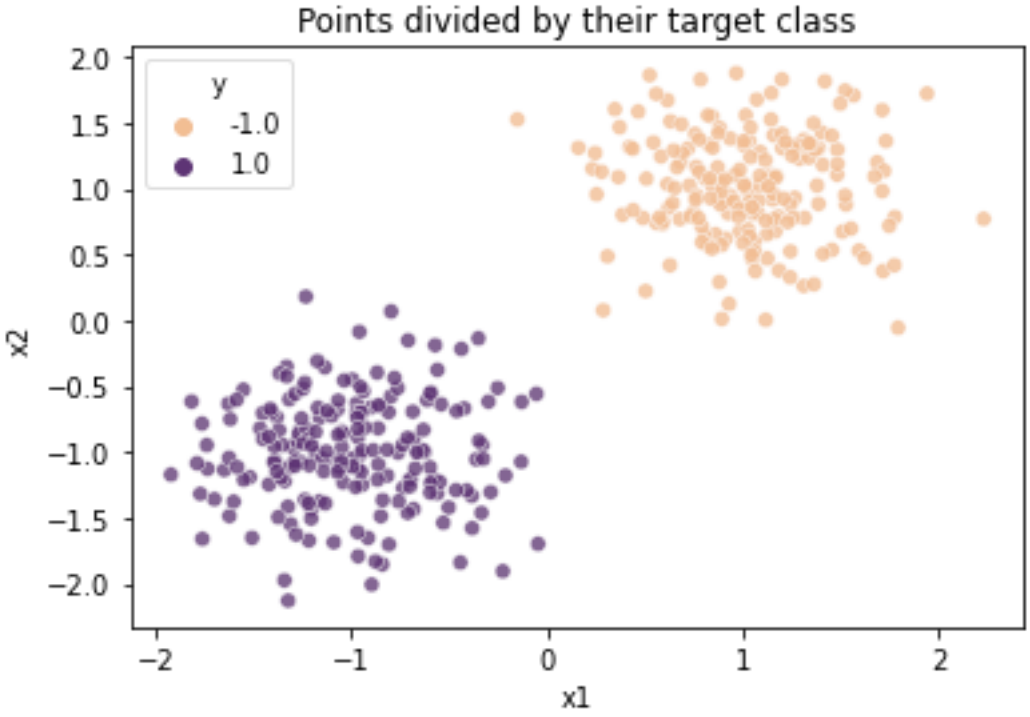
ج) فرایند کار این شبکه را از ورودی تا خروجی توضیح دهید.

طرز کار این شبکه به این شکل است که در ابتدا نور و صدا در قسمت z۲ ، منفجر شدن بمب اول را شناسایی می‌کنند که این قسمت نیز با کمک delay ای که در z۱ اتفاق می‌افتد امکان پذیر می‌باشد. با توجه به اینکه بمب دوم بعد از ۲ step زمانی منفجر می‌شود، زمانی که نورون z۲ خروجی ۱ می‌دهد نور مربوط به بمب دوم نیز شناسایی شده‌است. با ایجاد یک delay دیگر در قسمت z۳ برای در نظر گرفتن اینکه صدای بمب دوم نیز یک step زمانی بعد از نور آن قابل شناسایی خواهد بود، تمامی این اتفاقات یعنی نور با یک فاصله زمانی delay ، صدای شناسایی شده در لحظه و بمب شناسایی شده در

دو فاصله زمانی قبل‌تر، همگی جمع شده و AND آن‌ها به عنوان خروجی در نظر گرفته می‌شود تا زمانی که همه آن‌ها با هم اتفاق افتادند، شخص کلید را بزند.

سوال ۲ – Perceptron

الف) رسم داده‌ها در scatter plot گفته شده.

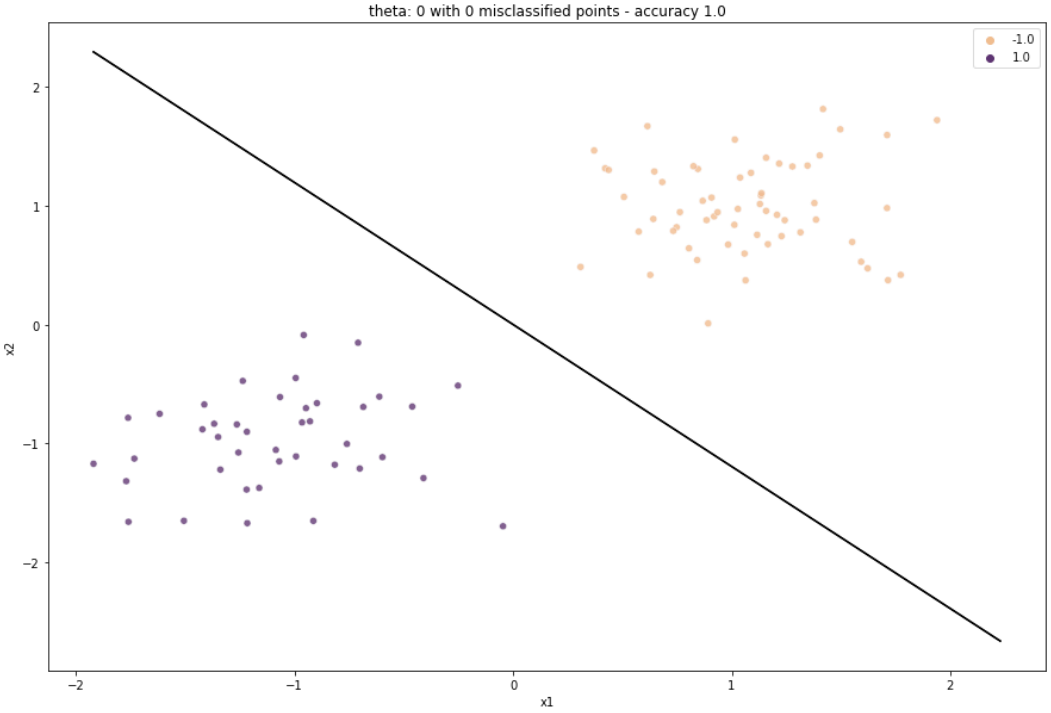


نقاط با توجه به کلاس آن‌ها با رنگ‌های متفاوت

ج) نتایج استفاده از نورون آموزش دیده روی داده‌های train برای پیشبینی کلاس مربوط به داده‌های تست.

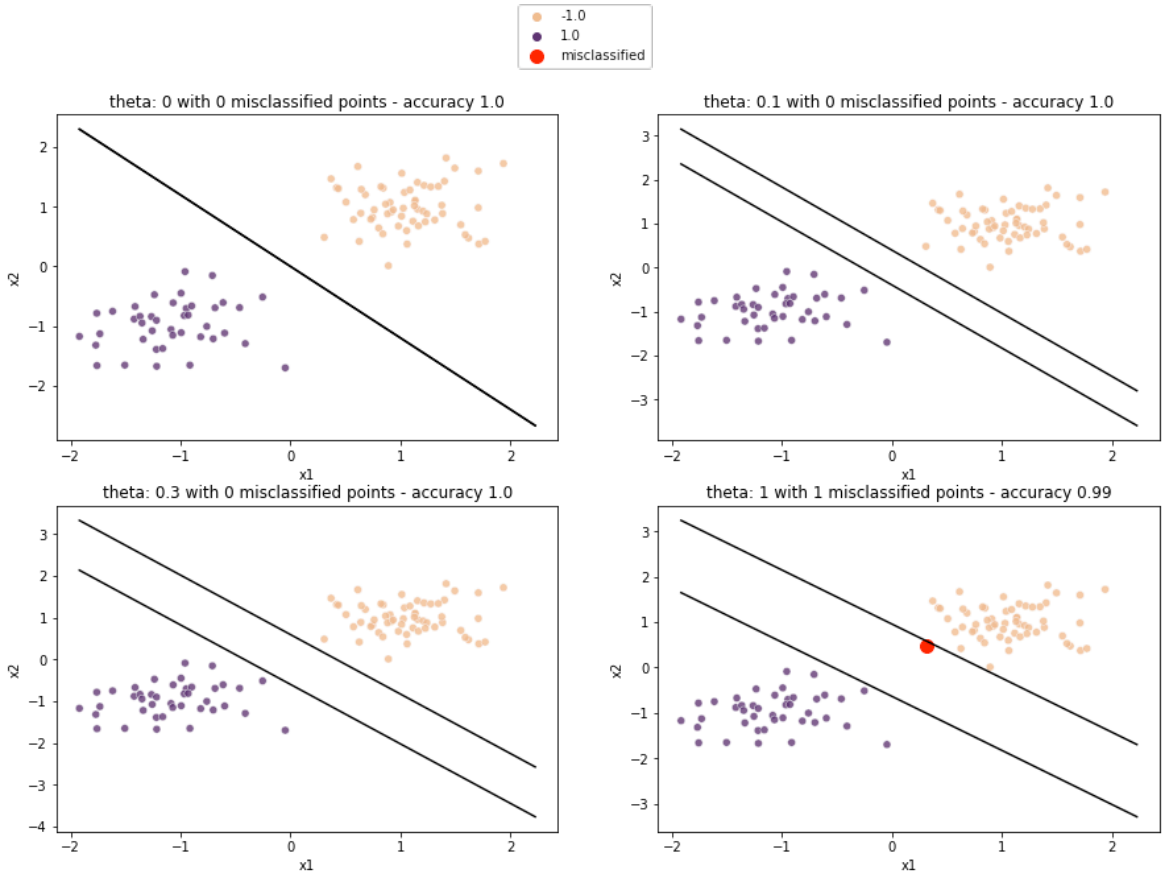
د) در این قسمت نتایج قسمت قبل را با استفاده از مقادیر متفاوت برای تتا بدست آوردیم.

نکته‌ای که باید به آن دقت کنید این است که اگر در فرآیند یادگیری تعداد epoch ها را بدون محدودیت در نظر بگیریم، حتی با بالا بردن تتا، هیچ نقطه‌ای داخل این دو خط نخواهد افتاد زیرا با در نظر گرفتن وزن‌های بیشتر، مدل همچنان سعی در کمینه کردن خطا و در نهایت صفر کردن آن دارد



نقاط متعلق به تفکیک کلاس و همینطور خط جداکننده (مربوط به داده تست)

که در نتیجه آن با بالا بردن تتا، از نقطه‌ای به بعد تغییری در خطوط ایجاد شده مشاهده نمی‌شود تا خطایی در این مورد اضافه نشود.



شکل مربوط به نقاط به تفکیک کلاس و خط جداساز با استفاده از تتاهای متفاوت استفاده شده در خط

همینطور در مورد تاثیری که پارامتر تتا در مدل دارد می‌توان گفت این پارامتر، باعث می‌شود خطوط تا جای ممکن از هم دور شوند که به بیان بهتر باعث می‌شود نقاط نزدیک به مرکز این طبقه بند در قسمت صفر قرار گرفته و در قسمت‌های یک و منفی یک قرار نگیرند و به این ترتیب تنها نقاطی که به طور کاملا مشخص در قسمت یک و منفی یک قرار می‌گیرند دارای کلاس یک و منفی یک خواهند شد. برای جمع بندی نیز می‌توان گفت میزان دقتی که مدل دارد به همین دلیل کمتر خواهد شد که حتی شاید به نوعی regularization term نیز به حساب بیاید.

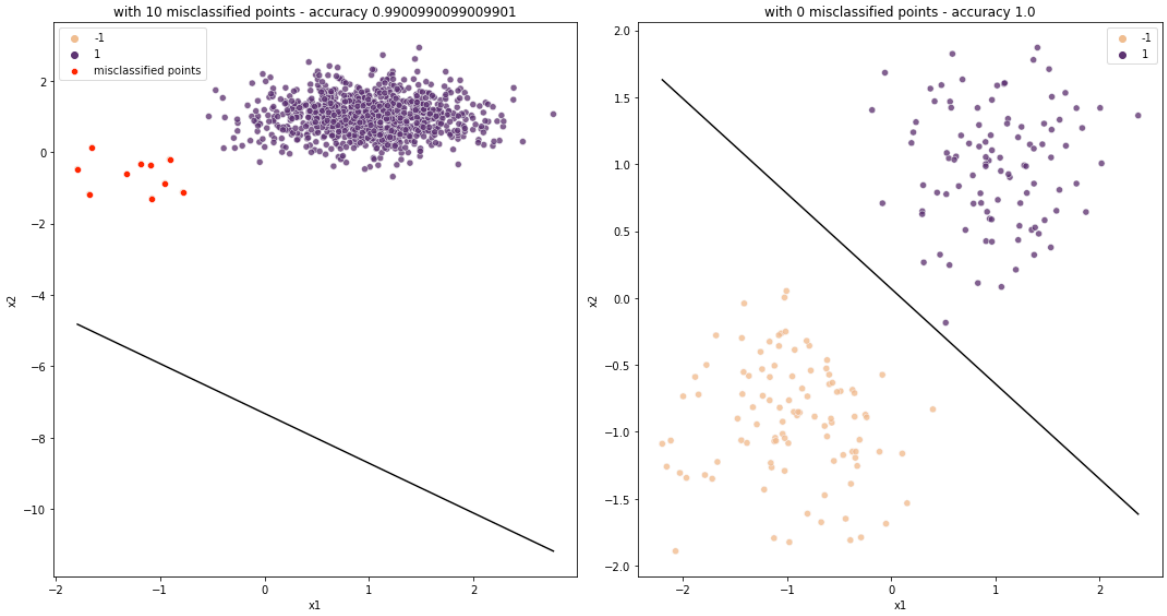
سوال ۳ – Adaline

الف) دو تفاوت و یک شباهت اصلی شبکه عصبی آدلاین و پرسپترون خطی را بیان کنید.

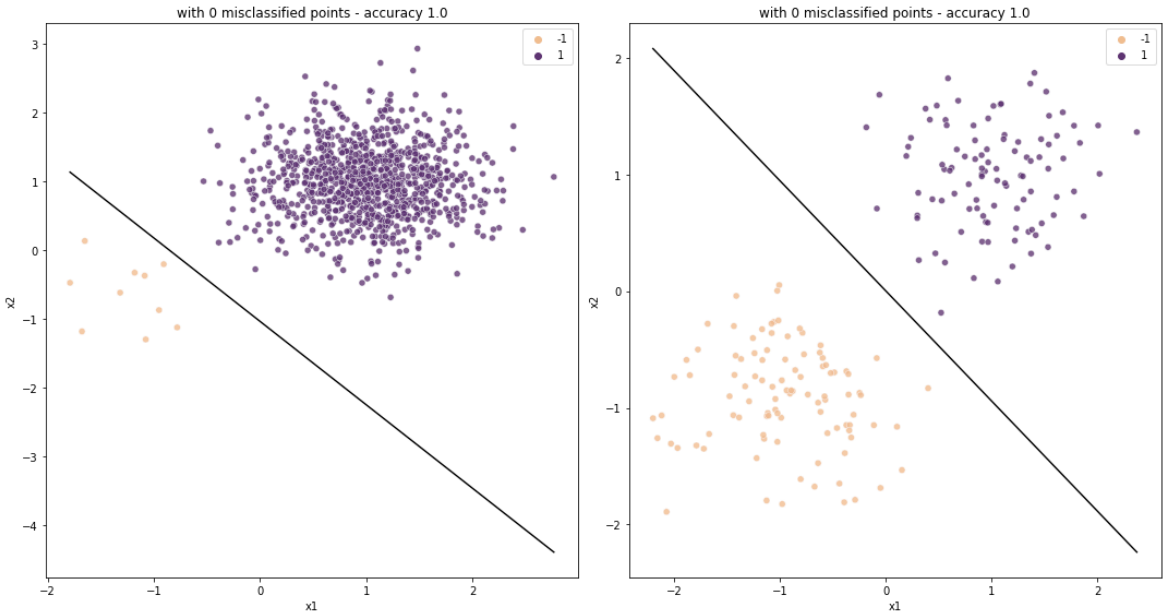
شباهت عمده‌ای که این دو شبکه با هم دارند مربوط به ساختار و architecture آن‌ها است که هردو کاملا شبیه هم هستند. تفاوت‌هایی که در مورد این دو شبکه موجود است یکی وجود و عدم وجود threshold است که در پرسپترون خطی وجود دارد و در نتیجه آن باعث بوجود آمدن دو خط می‌شود، در حالی که در adaline وجود ندارد و در نتیجه آن تنها یک خط جدا کننده ایجاد می‌کند. تفاوت دیگری که در مورد این دو مدل وجود دارد مربوط به قاعده‌های بروز رسانی وزن‌ها می‌باشد. البته این نکته را نیز در نظر بگیرید که اگرچه این دو قاعده با هم تفاوت دارند، اما چون علامت یکسانی دارند و در یک جهت حرکت می‌کنند، هر دو در جهت کم کردن خطا و ایجاد خط جدا کننده حرکت می‌کنند. همینطور در راستای تفاوت در قاعده بروزرسانی، تفاوت در شکل محاسبه مقدار ارور نیز وجود دارد که در حقیقت همین نکته نیز باعث تفاوت در قاعده بروزرسانی می‌شود.

ب) توضیح الگوریتم و رسم خط جدا کننده و همینطور داده‌های مربوطه.

الگوریتم به این شکل کار می‌کند که تا زمانی که خطاها صفر نشود داده‌های مربوط به train را به مدل نشان می‌دهیم. در هر کدام از این تکرارها، هر کدام از داده‌ها در ابتدا به مدل نشان داده و عمل forward pass را برای آن انجام می‌دهیم. قسمت forward pass نیز به این شکل انجام می‌شود که جمع وزن دار داده‌ها همراه با وزن‌های مربوطه محاسبه شده و با قسمت موسوم به bias جمع می‌شود. در انتها نیز برای بدست آوردن کلاس مربوطه علامت مقدار بدست آورده شده در نظر گرفته می‌شود. با توجه به اینکه مدل به درستی کار جداسازی را انجام داده یا خیر، خطا محاسبه شده و همینطور با استفاده از آن وزن‌ها و بایاس مربوط به بروز رسانی می‌شود. تمامی قسمت‌های گفته شده در مورد قسمت بعدی که از tanh استفاده می‌شود نیز به همین شکل بوده و تنها روش محاسبات تغییر پیدا خواهد کرد.



خط جداساز میان داده‌های دو کلاس انجام شده برای هر دو سری داده با استفاده از adaline و حالت عادی آن و تابع فعالساز غیر نرم



خط جداساز میان داده‌های دو کلاس انجام شده برای هر دو سری داده با استفاده از adaline بهبود یافته و تابع فعالساز نرم

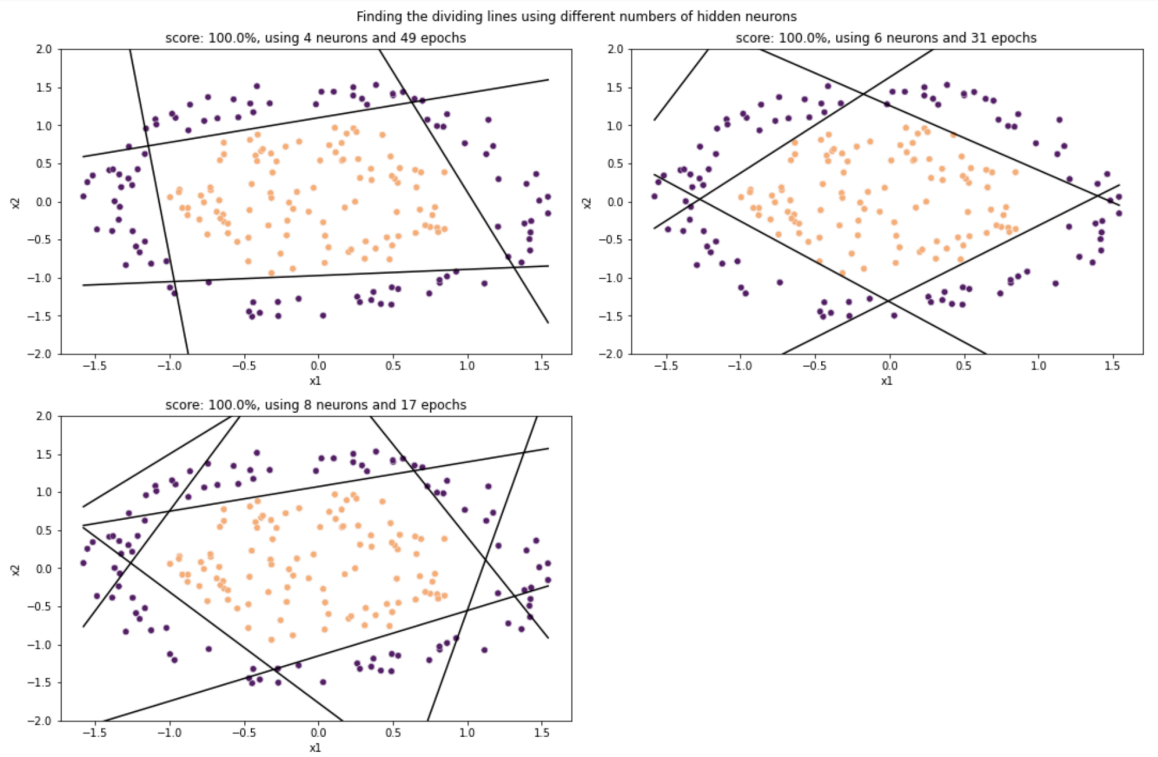
ج) تضمینی وجود ندارد که مدل در مواجهه با داده‌های نشان داده شده به خوبی عمل کند و یادگیری را انجام بدهد همانطور که در شکل‌های بالا نیز مشخص است. دلیل این امر این است که مقدار خطایی که این مدل سعی در کمینه کردن آن دارد کاملا دقیق نبوده و در مقدار خروجی مدل در محاسبه خطا، هم‌نوع با مقادیر درست کلاس‌ها نیست. همین امر باعث می‌شود تا اگر توزیع داده‌های مربوط به کلاس‌های متفاوت یکسان نباشد مدل نتواند یادگیری را به خوبی انجام دهد که در نتیجه آن خط جداساز نیز از دقت خوبی برخوردار نخواهد بود. همینطور می‌توان به این نکته نیز اشاره کرد که تابع

فعالساز نرمی در آن استفاده نشده که در نتیجه آن مقدار خطا و شیوه بروزرسانی نیز در تناسب با کلاس‌های خروجی تعیین نشده است.

برای رفع مشکل گفته شده می‌توان از تابع فعالساز نرمی استفاده کرد که با استفاده از آن هم بتوان تابع sign را شبیه‌سازی کرد و همینطور با کمک نرم بودن آن بتوان به درستی شیوه‌ای برای بروزرسانی با اثبات ریاضی ارائه داد. برای پیاده سازی نکات گفته شده از تابع tanh استفاده می‌کنیم که هم خواص مربوط به توابع خطی و مشتق پذیر را دارد و هم می‌توان با استفاده از آن یک تابع sign را شبیه سازی کرد. نتایج استفاده از این تابع در مدل در قسمت زیر آمده است که همانطور که مشهود است بهبود قابل توجهی مشاهده می‌شود. همینطور این نکته نیز وجود دارد که برای جمله خطا مقدار دقیق کلاس که هم‌نوع با خروجی است را در نظر می‌گیریم.

د) با در نظر گرفتن این که برای بروز رسانی از تابع نرمی استفاده می‌کنیم و به طور شهودی در هر بروز رسانی در جهت نزدیک‌تر شدن به یک دره حرکت می‌کنیم، تاثیر این پارامتر را می‌توان به یک شتاب در حرکت در جهت شیب داخل دره گفته شده تشبیه کرد. پس می‌توان گفت با زیاد بودن این پارامتر با شیب زیادی در جهت نزدیک شدن به دره حرکت می‌کنیم که در نتیجه آن ممکن است دره را رد کرده و به بدنه قسمت دیگر آن برسیم که در نتیجه آن مشکل convergence خواهیم داشت. همینطور با کم کردن این پارامتر نیز شیب ما کم خواهد بود که در نتیجه آن ممکن است نتوانیم از دره‌های بسیار کوچک در مسیر رسیدن به دره اصلی رد شویم و در نتیجه آن به sub-optimal solution خواهیم رسید. نمود نکات گفته شده در مسئله در دست نیز به این شکل است که اگر مقدار زیادی برای این پارامتر تعیین شود، ممکن است نتوانیم به جواب مورد نظر دست پیدا کنیم و خطا را به صفر برسانیم. همینطور اگر مقدار کمی برای این پارامتر در نظر بگیریم، این اتفاق می‌افتد که جواب دقیق را پیدا نمی‌کنیم و الگوریتم اگر چه فکر می‌کند به جواب رسیده ولی در حقیقت جواب دقیق درست نخواهد بود.

سوال ۴ – Madaline



خطوط بدست آمده با استفاده از madaline با ذکر epoch و همینطور دقت بدست آمده

تعداد epoch استفاده شده، تعداد نورون استفاده شده و همینطور دقت هر مدل را می‌توانید در شکل بالا مشاهده کنید. نکته‌ای که در مورد این اشکال وجود دارد این است که برای اینکه قالب مقایسه باشند، یک seed مشخص برای وزن‌ها و همینطور بایاس استفاده شده است. دیگر نکته‌ای که پر اهمیت می‌باشد این است که برای سادگی بیشتر، نورون انتهایی که برای مدل کردن and استفاده شده است به این شکل پیاده سازی شده است که اگر تمامی ورودی‌ها منفی یک باشند خروجی شبکه نیز منفی یک باشد و در غیر این صورت یک. خروجی منفی یک برای نقاط داخلی در نظر گرفته شده اند تا اگر این نقاط نسبت به تمامی خطها خروجی منفی یک می‌دهند خروجی شبکه نیز منفی یک باشد. دلیل این کار این می‌باشد که اگر به این شکل پیاده سازی را انجام دهیم که جمع وزن دار علامت خطها نسبت به نقطه را در نظر بگیریم ممکن است خطوط در بین نقاط بیرونی و داخلی قرار نگیرد ولی خروجی متناسب با یکی از این حالات را مشاهده کنیم که در نتیجه آن غلط می‌باشد. به عنوان مثال می‌توان ۴ خط نسبت به یک نقطه بالای آن و ۴ خط پایین آن باشند و همدیگر را خنثی کنند.

البته روش بروز رسانی مربوط به or می‌باشد با توجه به اینکه برای نقاط خارجی در نظر گرفته شده است. البته بروز رسانی این دو logic شبیه هم بوده و هر دو در یک راستا حرکت می‌کنند به این دلیل که قرار گرفتن نقاط داخلی در مرکز تمامی خطها در واقع این حالت را نیز ایجاد می‌کند که نقاط

خارجی هرکدام نسبت به حداقل یک خط بیرون قرار بگیرند. همینطور با دقت به تعداد epoch های طی شده برای رسیدن به نتیجه مد نظر می‌توان به این نتیجه رسید که برای تعداد نورون بالاتر، سرعت رسیدن به خطوط بهینه نیز بیشتر می‌باشد. اگر چه تعداد زیادی از این مشاهدات برای رسیدن به رابطه علیت و یا منطقی وجود ندارد اما می‌توان به طور شهودی به این نکته رسید که با تعداد نورون بیشتر رسیدن به خطوطی که بتوانند نقاط مورد نظر را در داخل خود قرار دهند راحت‌تر بوده، به این دلیل که با تعداد نورون بالاتر در حقیقت می‌توان این کار را کرد که هر خط به شکل relax تری به مکان مورد نظر خود برسد و به نوعی با تعداد خطوط بالا، دقت بالای مدل را پشتیبانی کنیم. این در حالی است که با تعداد خطوط کم هر خط می‌بایست از دقت بسیار بالایی برخوردار باشد. همینطور در بحث دقت نیز می‌توان به این شکل بحث کرد که با توجه به اینکه نقاط داده شده در تمامی مدل‌ها قابل رسیدن به دقت کامل هستند، با دادن زمان کافی به آن‌ها به دقت کامل خواهیم رسید در صورتی که اگر نقاط به شکلی بودند که این امکان وجود نداشت، دقت مدل‌هایی با تعداد نورون کمتر به طور قطع کمتر می‌بود.

نکته بسیار مهمی که وجود دارد این است. که تمامی نکات گفته شده در مورد نتایج گرفته شده کاملا وابسته به وضعیت اولیه خطوط دارند. با در نظر گرفتن این نکته، اینکه با چه تعداد epoch به حالت مورد نظر برسیم و یا اینکه با چه تعداد خط سریع‌تر به این حالت می‌رسیم کاملا تحت تاثیر قرار خواهد گرفت.