

سوال ۱

از آنجا که داده‌های mnist نرمالایزد اند. هنگامی که این داده‌ها را شافل کرده و اسلایس آنها را انتخاب کنیم داده‌های انتخابی نیز نرمالایزد خواهند بود.

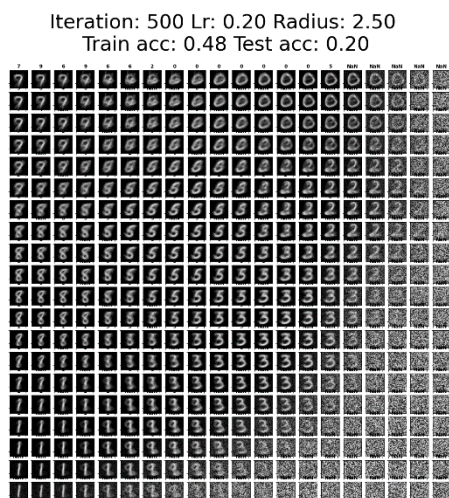
ثابت بودن نرخ یادگیری:

سرعت آپدیت کردن وزن‌ها در شروع کار و پایان آن یکی می‌باشد اما این کار درستی نیست چون هر چه به جلو می‌رویم وزن‌ها نزدیک حالت **Converge** میشود و آپدیت آن‌ها با نرخ بالا دوباره شبکه را از این حالت خارج می‌کند.

ثابت بودن شعاع همسایگی:

اگر داده‌ها در ابتدا دچار پیچ خوردگی و **Distortion** باشند این اثر تا آخر الگوریتم پیش می‌رود و شبکه شبیه شکل پروانه میشود و در مرکزش یک پیچ خوردگی بزرگ به وجود می‌آید. راه حل این مشکل این است که در ابتدا شعاع همسایگی بزرگ باشد و به مرور کاهش یابد.

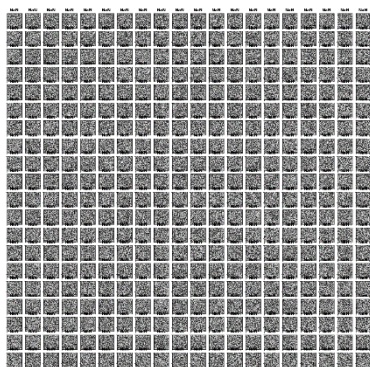
تصویر اعمال دو پدیده بالا روی داده‌ها با تعداد کمتری داده



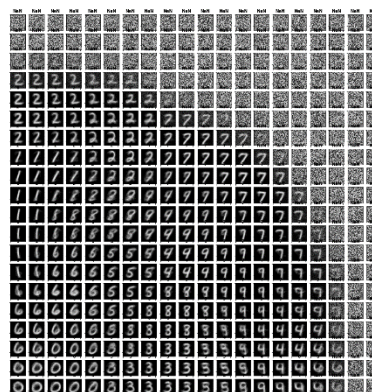
با افزودن تعداد epoch و تغییر مقادیر اولیه پارامترها نمایش کل داده‌ها امکان پذیر است که در اینجا به دلیل زمان بر بودن به همین مقدار اکتفا شده است.

برای مشاهده نتایج لازم است تا خط زیر در کد از حالت کامنت خارج گردد.

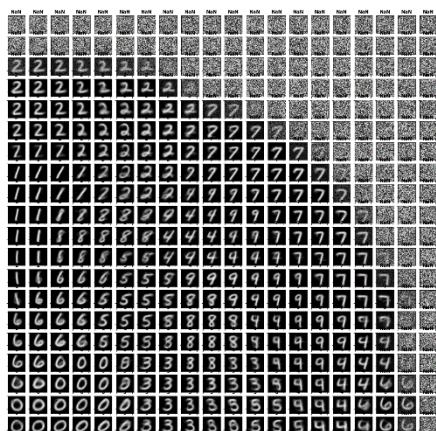
Iteration: 0 Lr: 0.20 Radius: 2.50
Train acc: 0.00 Test acc: 0.00



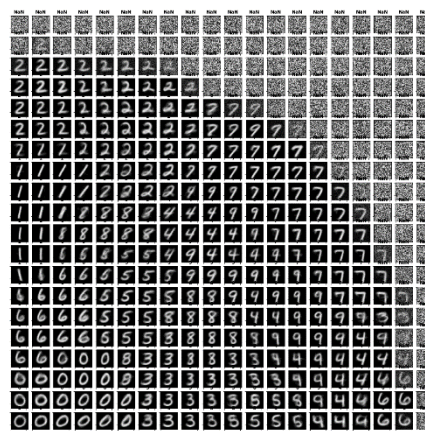
Iteration: 4500 Lr: 0.18 Radius: 1.52
Train acc: 0.67 Test acc: 0.64



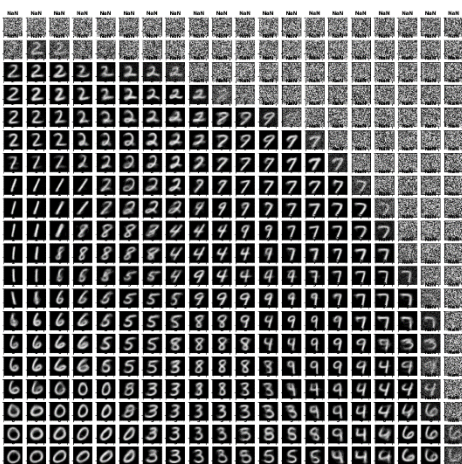
Iteration: 9000 Lr: 0.16 Radius: 0.92
Train acc: 0.81 Test acc: 0.78



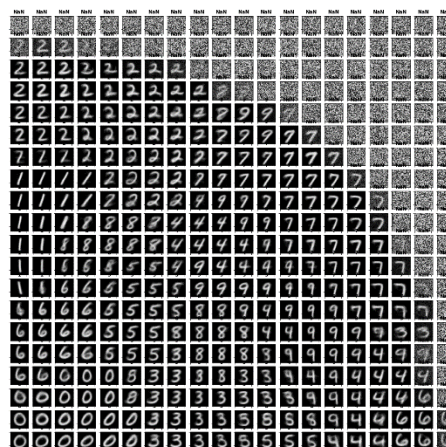
Iteration: 18000 Lr: 0.13 Radius: 0.50
Train acc: 0.89 Test acc: 0.88



Iteration: 31500 Lr: 0.09 Radius: 0.50
Train acc: 0.90 Test acc: 0.89



Train acc: 0.91 Test acc: 0.90



سوال ۲

نقشه استاندارد Kohonen از یادگیری بدون نظارت و لایه Kohonen منفرد استفاده می کند که امکان استفاده برای خوشه بندی و تجسم را فراهم می کند. از آنجا الگوریتم های بدون نظارت به برچسب نیاز ندارند این مدل نیز مستثنی نیست.

با تغییر ساختار این مدل می توان از آن برای دسته بندی استفاده کرد.

اولین تغییر، انتقال به یادگیری تحت نظارت با افزودن داده های ورودی در مورد خروجی های مورد نیاز است. اصلاح دوم، اجرای ساختار مدل سلسله مراتبی برای بهبود نتایج طبقه بندی است. توسعه سوم، اجرای مکانیزم بهینه سازی برای تنظیم پارامترهای مدل است، زیرا تعداد پارامترهای مدل افزایش یافته و تنظیم آنها دشوارتر است. در مرحله یادگیری، اولین تغییر در این واقعیت بود که پس از یافتن سلول برنده در لایه کوهونن، مدل طبقه بندی الگوهای آموزشی که به آن اختصاص داده شده بود را نیز ذخیره می کند. با کمک یک طبقه بندی شناخته شده از مجموعه آموزشی، می توانیم نرخ احتمال دسته های خروجی را در هر سلول تعیین کنیم که می تواند در تولید استفاده شود.

روش اول

هنگامی که داده های پیش بینی شده در فضای SOM را دارید، یک مجموعه داده دوبعدی دارید که باید به صورت خطی از هم جدا شوند (اگر دو کلاس به خوبی از هم متمایز شوند)، سپس می توانید از یک SVM یا رگرسیون لجستیک برای طبقه بندی این فضای جدید استفاده کنید. چیزی شبیه به افرادی که PCA را به عنوان مرحله پیش پردازش انجام می دهند.

روش دوم

با کد SOM اشتباه کنید، و به جای اختصاص دادن مقادیر وزن بر اساس ویژگی ها، می توانید سعی کنید و یک مقدار جدید را بر اساس برچسب اختصاص دهید، به این ترتیب، کلاستر را مجبور خواهید کرد که حول کلاس های شناخته شده خود متمرکز شود. هک است، اما می تواند کار کند.

سوال ۳

خوشه بندی روشی است برای گروه بندی مجموعه ای از نقاط داده به گونه ای که نقاط داده مشابه با هم گروه بندی شوند. بنابراین، الگوریتم های خوشه بندی به دنبال شباهت ها یا عدم شباهت ها در بین نقاط داده می گردند. خوشه بندی یک روش یادگیری بدون نظارت است، بنابراین هیچ برچسبی با نقاط داده مرتبط نیست. الگوریتم سعی می کند ساختار زیربنایی داده ها را پیدا کند

روش ها و الگوریتم های مختلفی برای انجام وظایف خوشه بندی وجود دارد که می توان آنها را به سه زیر دسته تقسیم کرد:

- خوشه بندی مبتنی بر پارتیشن: k-means, k-median
- خوشه بندی سلسله مراتبی: Agglomerative, Divisive
- خوشه بندی مبتنی بر چگالی: DBSCAN

DBSCAN

مخفف density-based spatial clustering of applications with noise است. می تواند خوشه های دلخواه شکل و خوشه های دارای نویز (یعنی نقاط پرت) را پیدا کند.

ایده اصلی پشت DBSCAN این است که یک نقطه به یک خوشه تعلق دارد اگر به نقاط زیادی از آن خوشه نزدیک باشد.

دو پارامتر کلیدی DBSCAN وجود دارد:

- **eps**: فاصله ای که محله ها را مشخص می کند. اگر فاصله بین آنها کمتر یا مساوی با **eps** باشد، دو نقطه همسایه در نظر گرفته می شوند.
- **minPts**: حداقل تعداد نقاط داده برای تعریف یک خوشه.

بر اساس این دو پارامتر، نقاط به عنوان نقطه مرکزی، نقطه مرزی یا نقطه پرت طبقه بندی می شوند:

- **نقطه مرکزی**: یک نقطه، یک نقطه هسته است اگر حداقل تعداد **minPts** نقاط (از جمله خود نقطه) در ناحیه اطراف آن با شعاع **eps** وجود داشته باشد.
- **نقطه مرزی**: نقطه ای که یک نقطه مرزی است که از یک نقطه مرکزی قابل دسترسی باشد و تعداد نقاط کمتری از تعداد **minPts** در محدوده اطراف آن وجود داشته باشد.

- نقطه پرت: نقطه ای نقطه پرت است اگر نقطه مرکزی نباشد و از هیچ نقطه اصلی قابل دسترسی نباشد.

تعاریف پارامترها و نقاط مختلف را یاد گرفته ایم. اکنون می توانیم در مورد نحوه عملکرد الگوریتم صحبت کنیم.

minPts و eps تعیین می شوند. یک نقطه شروع به طور تصادفی انتخاب می شود در منطقه همسایگی آن با استفاده از شعاع eps تعیین می شود. اگر حداقل تعداد minPts نقاط در همسایگی وجود داشته باشد، نقطه به عنوان نقطه مرکزی علامت گذاری می شود و تشکیل خوشه شروع می شود. اگر نه، نقطه به عنوان نویز مشخص می شود. هنگامی که یک خوشه شروع می شود (مثلاً خوشه A)، تمام نقاط در همسایگی نقطه اولیه جزء خوشه A می شوند. اگر این نقاط جدید نیز نقاط اصلی باشند، نقاطی که در همسایگی آنها قرار دارند نیز به خوشه A اضافه می شوند. مرحله بعدی انتخاب تصادفی نقطه دیگری از بین نقاطی است که در مراحل قبل بازدید نشده است. سپس همان رویه اعمال می شود. این فرآیند با بازدید از تمام نقاط به پایان می رسد.

با اعمال این مراحل، الگوریتم DBSCAN قادر است مناطق با چگالی بالا را پیدا کرده و آنها را از مناطق کم تراکم جدا کند.

K-means

خوشه بندی K-means یکی از ساده ترین و محبوب ترین الگوریتم های یادگیری ماشینی بدون نظارت است. به طور معمول، الگوریتم های بدون نظارت تنها با استفاده از بردارهای ورودی، بدون ارجاع به نتایج شناخته شده یا برچسب گذاری شده، از مجموعه داده ها استنتاج می کنند.

هدف K-means ساده است: نقاط داده مشابه را با هم گروه بندی کنید و الگوهای زیربنایی را کشف کنید. برای دستیابی به این هدف، K-means به دنبال تعداد ثابت (k) خوشه در یک مجموعه داده می گردد.

خوشه به مجموعه ای از نقاط داده اشاره دارد که به دلیل شباهت های خاص با هم جمع شده اند.

شما یک عدد هدف k را تعریف خواهید کرد که به تعداد مراکز مورد نیاز شما در مجموعه داده اشاره دارد. مرکز یک مکان خیالی یا واقعی است که مرکز خوشه را نشان می دهد. هر نقطه داده با کاهش مجموع مربع های درون خوشه ای به هر یک از خوشه ها اختصاص می یابد. به عبارت دیگر، الگوریتم K تعداد مراکز را شناسایی می کند و سپس هر نقطه داده را به نزدیکترین خوشه اختصاص می دهد، در حالی که مراکزها را تا حد امکان کوچک نگه می دارد.

means در K-means به میانگین گیری داده ها اشاره دارد. یعنی یافتن مرکز.

برای پردازش داده های یادگیری، الگوریتم K-means در داده کاوی با اولین گروه از مرکزهای به طور تصادفی انتخاب شده شروع می شود، که به عنوان نقطه شروع برای هر خوشه استفاده می شود، و سپس محاسبات تکراری را برای بهینه سازی موقعیت مرکزها انجام می دهد.

ایجاد و بهینه سازی خوشه ها را در موارد زیر متوقف می کند:

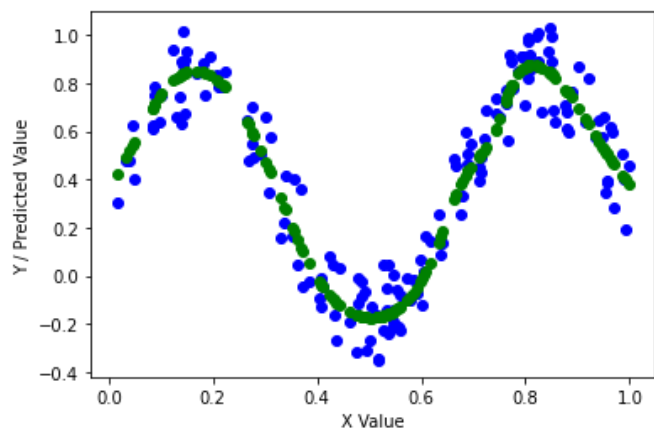
مرکزها تثبیت شده اند - هیچ تغییری در مقادیر آنها وجود ندارد زیرا خوشه بندی موفقیت آمیز بوده است.

تعداد تکرارهای تعریف شده به دست آمده است.

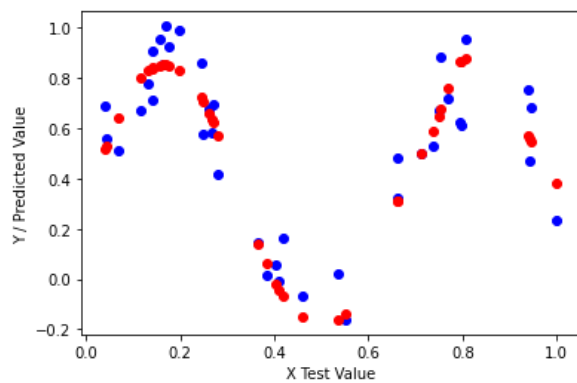
سوال ۴

۱.۴

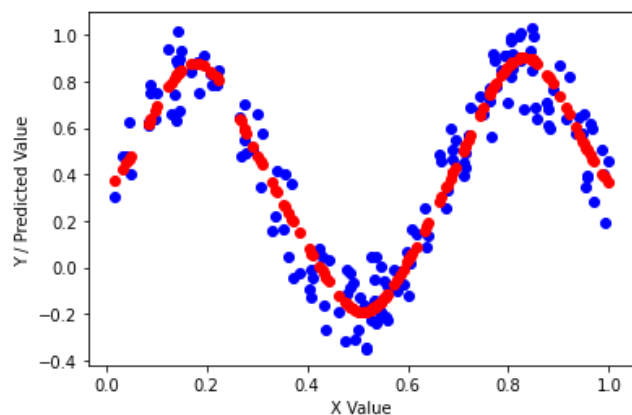
تصاویر خروجی تابع و پیش بینی روی داده train



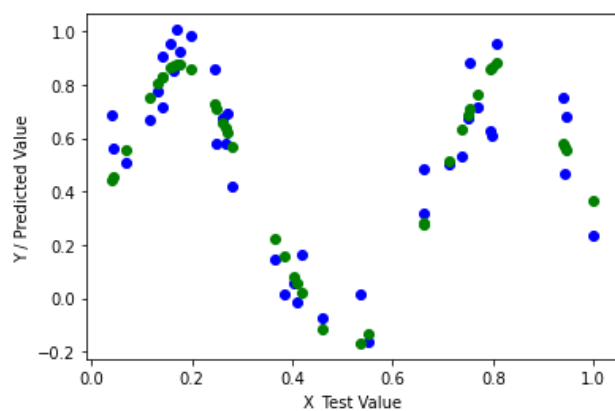
تصاویر خروجی تابع و پیش بینی روی داده test



۲.۴



تصاویر خروجی تابع و پیش بینی روی داده train



تصاویر خروجی تابع و پیش بینی روی داده test

۳.۴

همانطور که مشاهده می‌شود مدل RBF انطباق بیشتری روی تابع سینوس دارد نتیجه RBF بسیار دقیق تر است زیرا برای توابع غیر خطی و Classification های غیر خطی مدل های Radial به دلیل اینکه از متد های گاوسی استفاده میکنند، عملکرد بهتری دارند .