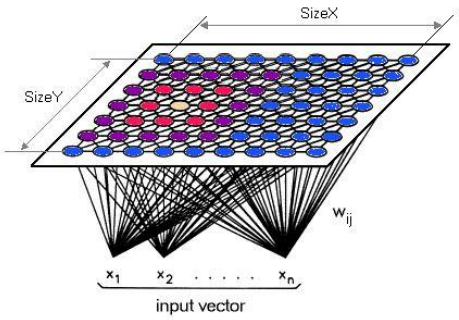
علیرضا خیاطیان

۱۴۰۰/۸/۲۸

تمرین دوم درس شبکه های عصبی

شبکه خود سازمانده

شبکه خودساز مانده یک شبکه دو لایه است که لایه اول دارای تعداد نورون متناسب با ابعاد داده ورودی است و لایه خروجی یک لایه است که میتوان آنرا به صورت خطی، دوبعدی و یا چند بعدی نمایش داد که به منظور یادگیری بدون نظارت و برای کارهایی مثل دسته بندی، کاهش ابعاد داده، بصری سازی اطلاعات و ...میتوان به کار برد.



شکل ۱: شبکه خودسازمانده کوهونن

در این شبکه طی یک فرآیند یادگیری رقابتی، نورون برنده مشخص می‌شود و وزن نورون بنده و نورون های همسایه آن، بروز رسانی می‌شوند. در ابتدای فرآیند سعی بر این است که شعاع همسایگی زیاد در نظر گرفته شود و در دفعات بعدی این شعاع با یک تابع خطی و یا گوسی و ... کاهش یابد. همچنین برای نرخ یادگیری نیز از همین قانون استفاده می‌کنیم و در ابتدا نرخ بالایی را در نظر می‌گیریم و سپس آن را در هر دفعه کاهش بر اساس یک رابطه خطی و یا یک رابطه گوسی کاهش می‌دهیم. رابطه زیر برای به روز رسانی وزن ها در نظر گرفته می‌شود.

$$W_j' = W_j + \beta NS[X - W_j] \qquad (1)$$

که در رابطه فوق NS تابع همسایگی و β نرخ یادگیری است که در زیر نمونه‌های خطی این دو نشان داده شده است.

$$\beta_t = \beta_\circ(1 - \frac{t}{T})$$

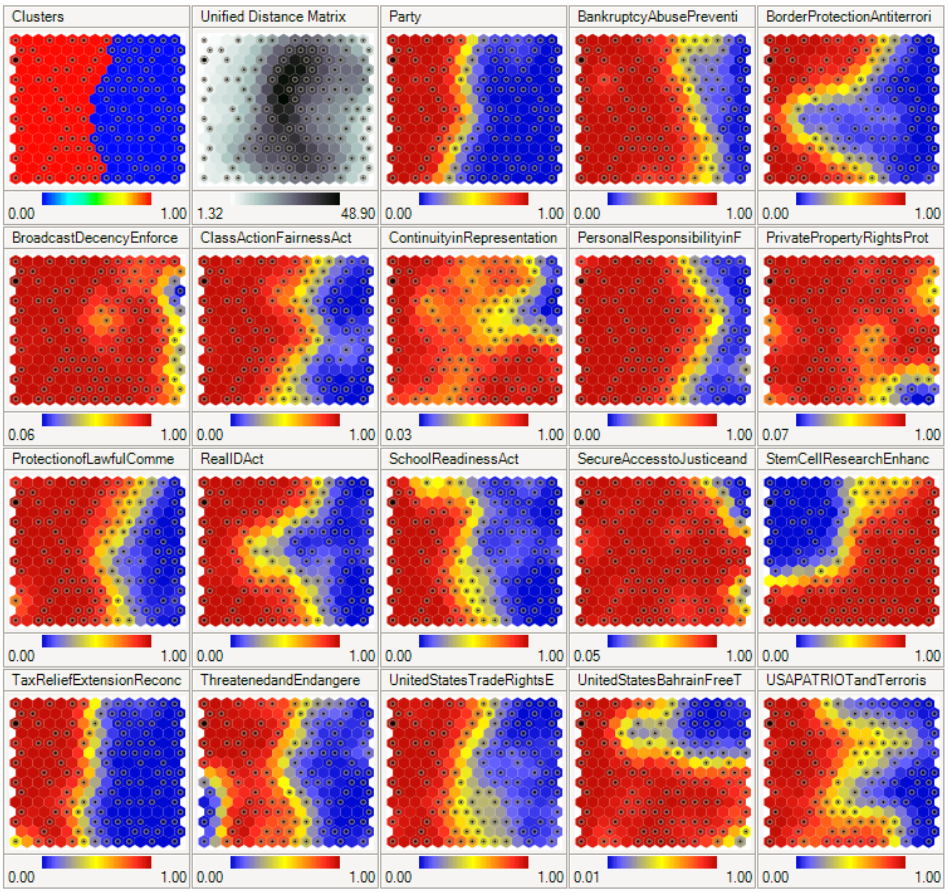
$$NS = e^{\frac{d_{ij}^2}{2\sigma^2}}$$

$$\sigma_t = \sigma_\circ(1 - \frac{t}{T})$$

در رابطه فوق T تعداد دفعات تکرار است.

یکی از ویژگی های شبکه SOM این است که داده‌های با ابعاد بالا را می‌توان به یک محور یک بعدی و یا یک صفحه دوبعدی و یا سه بعدی به وسیله یک نگاشت غیرخطی تصویر نمود .از این رو یکی از کاربردهای این شبکه کاهش ابعاد داده است که از این طریق می‌توان تحلیل خوبی از داده های با حجم بالا به وسیله بصری سازی پیدا کرد.

SOM یک الگوریتم خوشه‌بندی بدون نظارت است. به این صورت که نمونه‌های مشابه را در نقشه ویژگی نزدیک‌تر نشان می‌دهد (یعنی نمونه‌های مشابه را به گره‌هایی که به هم نزدیک‌تر هستند نگاشت می‌کند). بنابراین آنچه که SOM در واقع تولید می‌کند، نگاشتی از فضای ورودی X به فضای کاهش‌یافته Y است (متداول‌ترین آنها یک شبکه دو بعدی است که Y را یک فضای دو بعدی می‌کند). حال از این نقشه نقشه های ویژگی به دست آمده می‌توانیم برای تحلیل و بصری سازی داده ها استفاده کنیم.



شکل ۲: نقشه‌های ویژگی ارزیابی آرا نمایندگان کنگره آمریکا

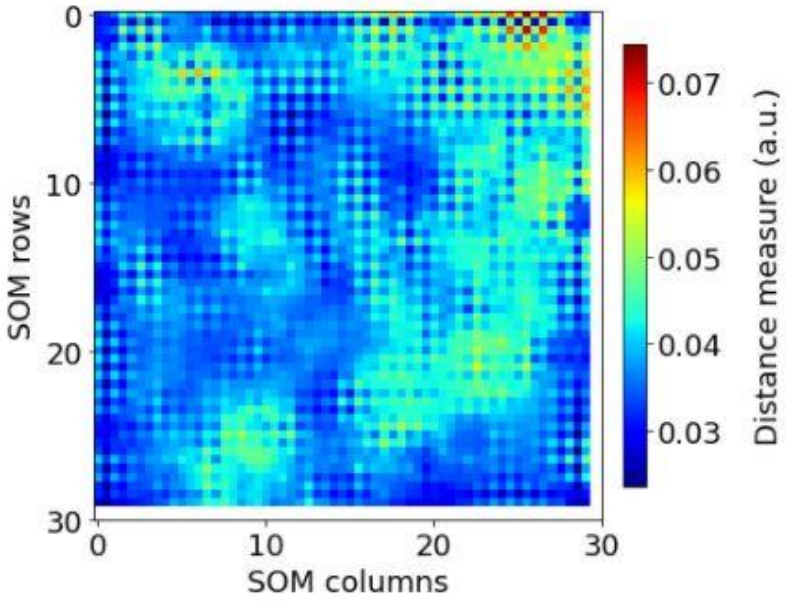
در شکل بالا شبک خود سازمانده روی مجموعه داده رای های مثبت و منفی نمایندگان کنگره به لایحه‌های مختلف نمایش داده شده است. نمودار اول خوشه بندی را هنگامی که داده ها به دو خوشه تقسیم می شوند نشان می دهد. نمودار دوم میانگین فاصله را با همسایگان نشان می دهد: فواصل بزرگتر تاریک تر هستند. طرح سوم عضویت حزب جمهوری خواه (قرمز) یا دموکرات (آبی) را پیش بینی می کند. نمودارهای دیگر هر کدام نقشه حاصل را با مقادیر پیش‌بینی‌شده در یک بعد ورودی پوشش می‌دهند: قرمز به معنای رای «بله» پیش‌بینی‌شده در آن لایحه، آبی به معنای رأی «نه» است. در واقع می‌توان بررسی کرد که هر طرح به چه میزانی در جدا کردن نمایندگان دو حزب از هم تاثیر دارد.

آماده سازی مجموعه داده‌ها

مجموعه داده‌ها شامل ۸۰۰۰۰ نمونه با ابعاد ۴۷۲۳۶ می‌باشد. هدف ما از این تمرین استفاده از شبکه خودسازمانده کوهونن برای کاهش ابعاد داده ها و بصری سازی آن است. فرم داده ها sparse matrix است که تنها درایه های غیر صفر را ذخیره می‌کند و بخش زیادی از مجموعه داده ها صفر است. برای اعمال som روی مجموعه داده ها نیاز است که آن ها را به numpy array تبدیل کنیم که به این ترتیب علاوه بر مقادیر غیر صفر باید حجم زیادی صفر نیز اضافه شود. به دلیل اینکه حجم RAM در colab جوابگوی پردازش این حجم از دیتا نیست ما به کمک resample از میانه نمونه های داده شده بخش از آن ها را انتخاب می‌کنیم.

SUSI

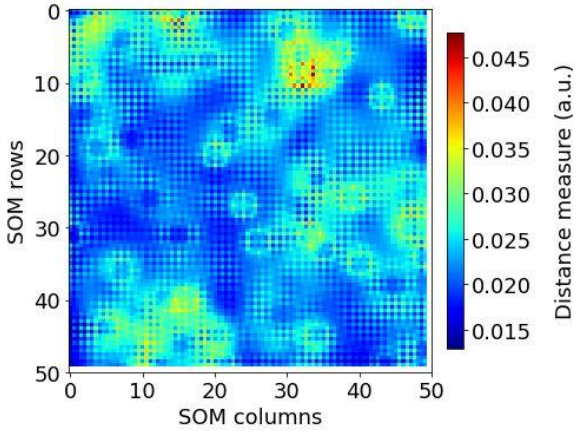
برای اعمال som از کتابخانه susi استفاده می‌کنیم. این کتابخانه به نسبت سایر کتابخانه ها همچون minisom عملکرد مناسب تری دارد و برای اجرا به RAM و زمان کمتری نیاز دارد. پس از نصب این کتابخانه، شبکه som را با ابعاد 30*30*47236 ایجاد می‌کنیم. تعداد تکرار های را برابر ۱۰۰۰ و نرم یادگیری اولیه را ۰,۷ در نظر می‌گیریم. برای اعمال همسایگی نیز از تابع گوسی استفاده می‌کنیم.در شکل زیر نمودار u-matrix را مشاهده می‌کنید.



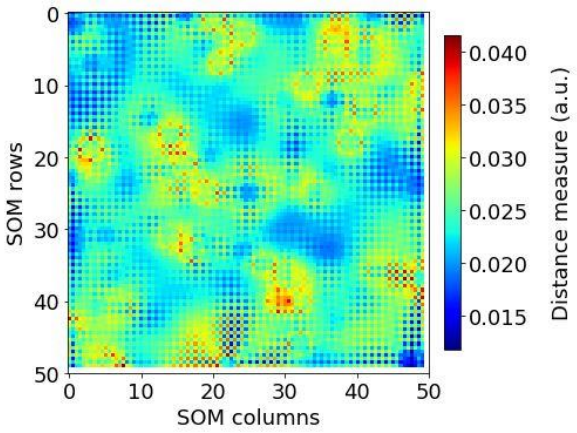
شکل ۳: نقشه u-matrix برای SOM 30*30

در شکل بالا نورون های آبی نزدیک به هم هر کدام یک منطقه را تشکیل می‌دهند. در واقع ناحیه های زرد رنگ را می‌توان مرز های جداکننده قسمت های مختلف در نظر گرفت. نورون‌های آبی رنگ نشان دهنده فاصل کم بین بردار وزن‌های این نورون‌ها با نورون‌های همسایه آن است از این موضوع می‌توان نتیجه گرفت تراکم داده‌ها در قسمت آبی رنگ بیشتر است. به بیان دیگر شبکه SOM یک تصویر از داده ها می‌دهد که داده های نزدیک به هم را به نورون های مجاور آبی رنگ تصویر می‌کند. و داده های با فاصله زیاد را به نورون های متمایل به قرمز تصویر می‌کند. دقت شود حتی اگر داده ها به نورون های مجاور تصویر شوند اما به فرض یک نورون آبی و دیگری قرمز باشد. نشان دهنده فاصله زیاد داده ها از هم می‌باشد. در واقع با این نحوه رنگ آمیزی، شبکه SOM چگالی توزیع داده ها را به صورت گسسته به نمایش می‌گذارد.

در بررسی یک نقشه دیگر ، شبکه som را با ابعاد 50*50*47236 ایجاد می‌کنیم. تعداد تکرار های را برابر ۱۰۰۰ و نرم یادگیری اولیه را ۰,۷ در نظر می‌گیریم. و این بار ۷۰۰۰ نمونه را به صورت تصادفی انتخاب می‌کنیم تا RAM داده شده توسط colab جواب گوی حجم محاسبات باشد. . برای اعمال همسایگی نیز از تابع گوسی استفاده می‌کنیم. همچنین، این نقشه را بار دیگر با نرخ یادگیری اولیه ۰,۵ به فرض ثابت بودن سایر پارامتر ها اجرا کردیم. در شکل زیر نمودارهای u-matrix را مشاهده می‌کنید.



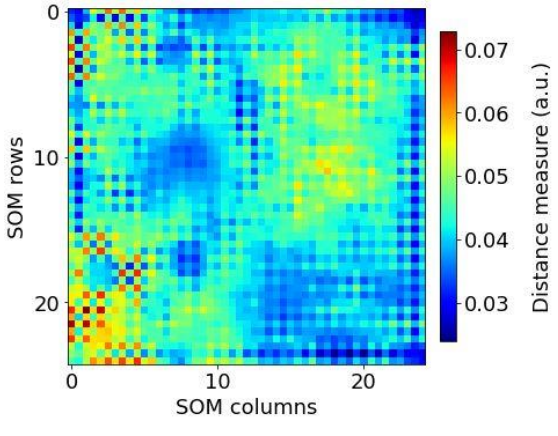
شکل ۴: نقشه u-matrix برای SOM 50*50 با نرخ یادگیری اولیه 0.5



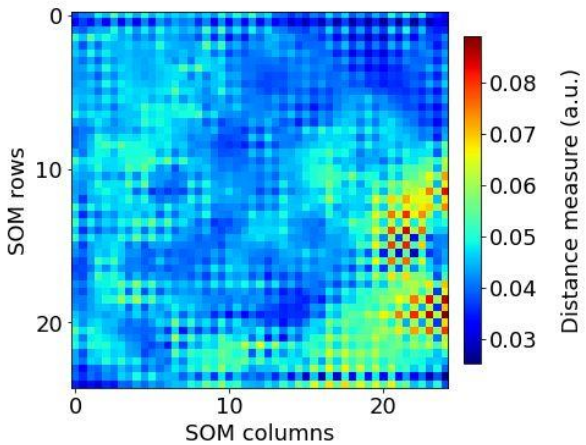
شکل ۵: نقشه u-matrix برای SOM 50*50 با نرخ یادگیری اولیه 0.7

همان طور که مشاهده می‌کنید نرخ یادگیری اولیه تاثیر زیادی درعملکرد الگوریتم دارد. و اجرای الگوریتم با نرخ ۰,۵ باعث متراکم شدن نقشه شده است. در شکل ۳ به خوبی می‌توان داده های تصویر شده به نود های مجاور آبی رنگ دسته بندی کرد و از نود های زرد رنگ نیز به عنوان مرز استفاده کرد.

همچنین در زیر، شبکه som را با ابعاد 25*25*47236 ایجاد می‌کنیم. تعداد تکرار های را برابر ۱۰۰۰ و نرخ یادگیری اولیه را ۰,۷ و .۰۵در نظر می‌گیریم. و این بار ۱۲۰۰۰ نمونه را به صورت تصادفی انتخاب می‌کنیم تا RAM داده شده توسط colab جواب گوی حجم محاسبات باشد. . برای اعمال همسایگی نیز از تابع گوسی استفاده می‌کنیم.در شکل زیر نمودار u-matrix را مشاهده می‌کنید.



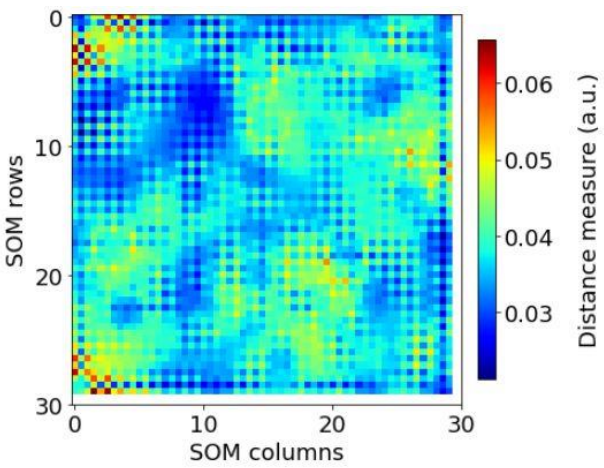
شکل ۶: نقشه u-matrix برای SOM 25*25 با نرخ یادگیری اولیه 0.5



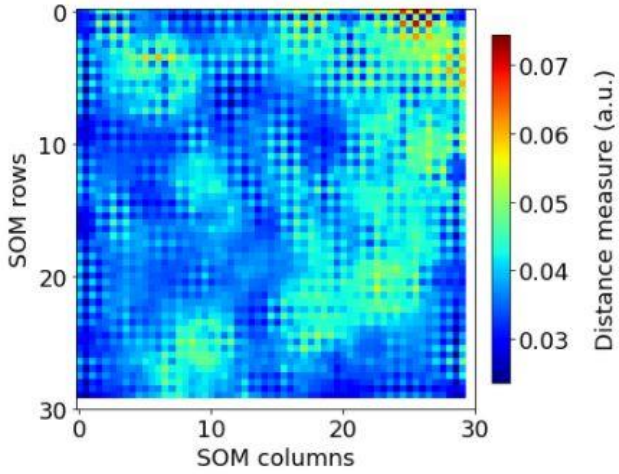
شکل ۷: نقشه u-matrix برای SOM 25*25 با نرخ یادگیری اولیه 0.7

برای نقشه 25*25 بر خلاف نقشه 50*50 نرخ یادگیری 0.7 عملکرد بهتری نسبت به مقدار 0.5 دارد.

همچنین در زیر، شبکه som را با ابعاد 47236*30*30 ایجاد می‌کنیم. تعداد تکرار های را برابر ۱۰۰۰ و نرخ یادگیری اولیه را ۰,۷ و ۰.۵در نظر می‌گیریم. و این بار ۱۰۰۰۰ نمونه را به صورت تصادفی انتخاب می‌کنیم. برای اعمال همسایگی نیز از تابع گوسی استفاده می‌کنیم.در شکل زیر نمودار u-matrix را مشاهده می‌کنید.



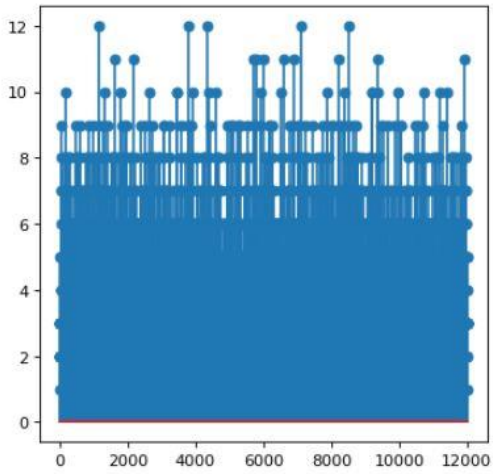
شکل ۸: نقشه u-matrix برای SOM 30*30 با نرخ یادگیری اولیه 0.5



شکل ۹: نقشه u-matrix برای SOM 30*30 با نرخ یادگیری اولیه ۷0.

خوشه بندی

مجموعه داده های داده شده multi-label هستند و هر داده به چند کلاس از ۱۰۳ کلاس موجود متعلق است. به عنوان مثال ممکن است یک خبر یا یک داستان هم در کلاس موضوعات فرهنگی باشد و هم در کلاس موضوعات بین المللی. در زیر نموداری از تعداد کلاس هایی که هر داده به آن متعلق است آمده است.

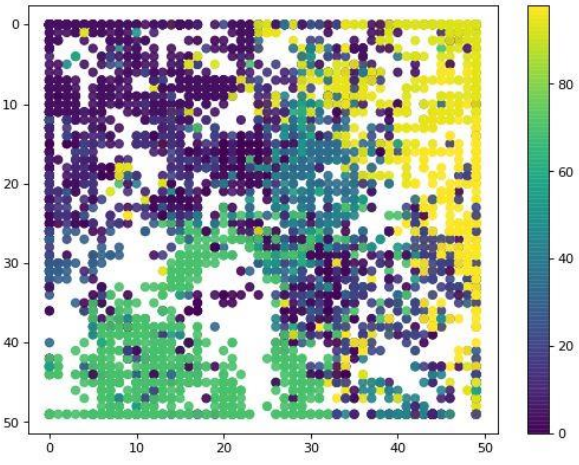


شکل ۱۰: نمایش تعداد کلاس های هر نمونه

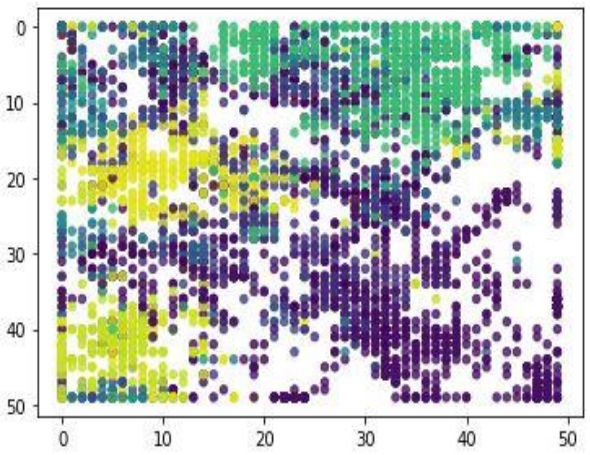
برای خوشه بندی از کتابخانه susi استفاده کردیم. تابع get_cluster در این کتابخانه به این صورت عمل می‌کند که هر داده را باتوجه به تنسور های به وجود آمده در نقشه som هر یک از ورودی ها را به یک نورون از نقشه نگاشت می‌کند بر مبنای ترتیب داده های داده شده نورون های مورد نظر را به عنوان خروجی می‌دهد. در واقع هر نورون را را یک خوشه درنظر می‌گیریم. به کمک som به دست آمده در قسمت قبل داده ها را به نورون، خوشه‌ها، نگاشت می‌کنیم. برای نمایش خوشه بندی و رنگ آمیزی خوشه ها به کمک label ها باید در نظر داشته باشیم که داده ها multi-label هستند و نمایش تمام label های داده ممکن نیست. ما برای نمایش خوشه‌ها با label ها آمدیم و برای هر داده اولین label داده را در نظر گرفتیم و به کمک scatter مجموع آن ها را نمایش دادیم. در زیر نمایش های خوشه بندی som های با اولین label داده ها در بخش قبل آورده شده است.

تحلیل توزیع داده‌ها

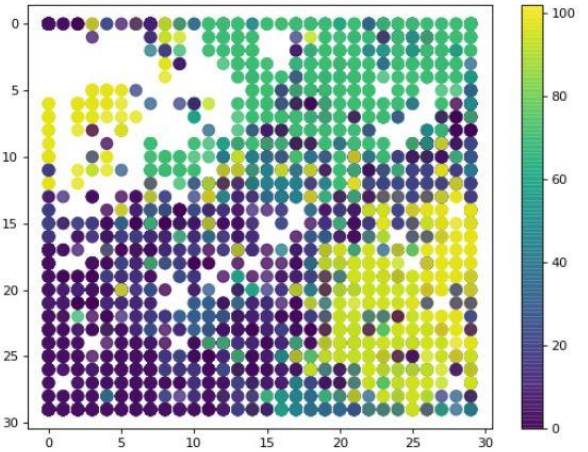
همان طور که مشاهده می‌کنید، عمل خوشه بندی به خوبی انجام شده و داده‌ها با ابعاد بسیار بالا به خوبی توسط som تفکیک شده است. حال به تحلیل نمودار نقشه som 30*30 با نرخ یادگیری ۰,۷ که نمودار u-matrix آن نسبت به بقیه حالت ها داده هار را متراکم تر کرده است می‌پردازیم. همان طور که از همه شکل ها از جمله شکل ۱۴ که مورد نظر ماست آشکار است می‌توان با 4 خوشه داده ها را خوشه بندی کرد که با 4 رنگ بنفش، آبی، سبز و زرد نمایش داده شده است. در رابطه با جدایی پذیری داده ها در شکل ۱۴، تا حد قابل قبولی جدایی پذیر هستند به جز داده خوشه آبی رنگ که با خوشه سبز رنگ در بخش هایی هم پوشانی دارند و جداپذیر نیست. برای پیچیدگی داده ها می‌توان گفت با توجه به ابعاد بسیار بالای داده ها، نقشه شکل نسبتا ساده ایی از داده ها داده است و می‌توان گفت داده ها پیچیدگی زیادی همچون اشکال مارپیچی و XOR ندارند و با اعمال یک MLP به خروجی som آن ها را می‌توان دسته بندی کرد. داده های نویزی به داده‌هایی گفته می‌شود که در هیچ خوشه ایی جای نگیرند. با توجه به شکل ۱۴ می‌توان گفت داده‌ها دارای نویز کمی هستند.



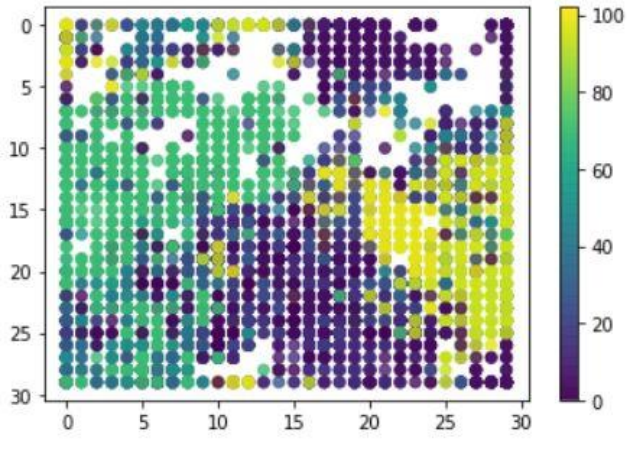
شکل ۱۱: نقشه SOM 50*50 با نرخ یادگیری اولیه 0.5



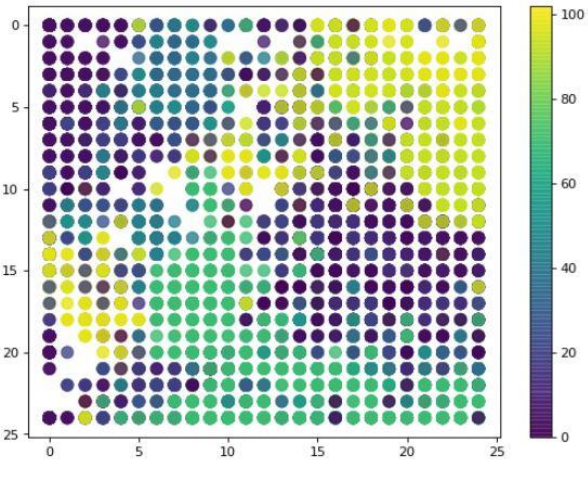
شکل ۱۲: نقشه SOM 50*50 با نرخ یادگیری اولیه 0.7



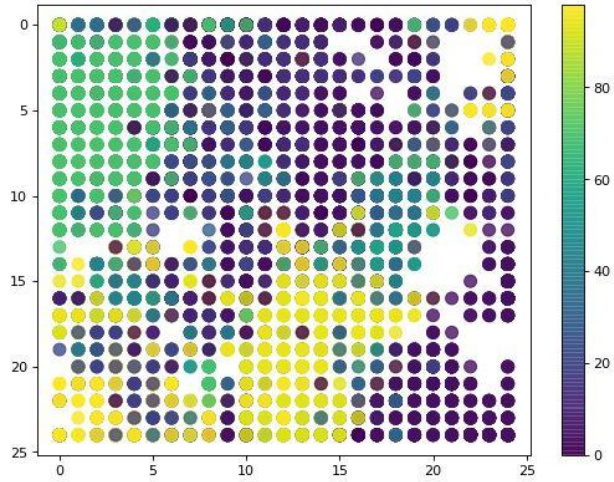
شکل ۱۳: نقشه SOM 30*30 با نرخ یادگیری اولیه 0.5



شکل ۱۴: نقشه SOM 30*30 با نرخ یادگیری اولیه 0.7



شکل ۱۵: نقشه SOM 25*25 با نرخ یادگیری اولیه 0.5



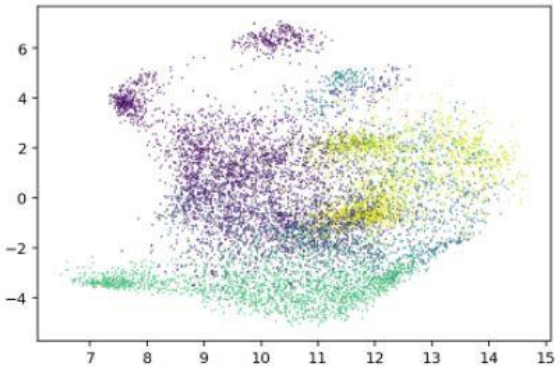
شکل ۱۵: نقشه SOM 25*25 با نرخ یادگیری اولیه 0.7

Purity

به طور کلی purity یک معیار ارزیابی مناسب برای داده های تک label می‌باشد و معیارهای ارزیابی دیگری برای داده هاس multi-label مورد استفاده می‌گردد. Purity را به این صورت محاسبه می‌کنیم: ابتدا یک تنسور با ابعاد نقشه و با عمق کلاس‌ها یا برچسب‌ها در نظر می‌گیریم. برای هر نود، داده‌هایی که به آن نگاشت شده را پیدا می‌کنیم. حال با بررسی کلاس‌های داده های نگاشت شده به آن خانه بردار تنسور هر خانه را به ازای هر برچسب ان داده یکی اضافه می‌کنیم. در نهایت نیز کلاسی که دارای بیشترین مقدار است را مرجع در نظر می‌گیریم. با جمع کلاس های مرجع هر نود و تقسیم آن ها به کل داده‌ها مقدار purity را به دست می‌آوریم. مقدار purity با این روش برای som 25*25 و نرخ یادگیری ۰,۷ برابر ۸۸,۸۱٪ شد. در روش دیگر برای محاسبه purity همچون کشیدن نقشه های بخش قبل از اولین کلاس هر داده نگاشت شده اسفاده می‌شود که در این روش purity برابر ۶۲,۷۵٪ می‌شود. همچنین برای som 50*50 نیز purity برای دو روش فوق به ترتیب برابر ۹۱,۹۵٪ و ۷۱,۷۱٪ می‌باشد.دقت شود مقایسه purity برای som ها با ابعاد متفاوت کار درستی نیست.

لازم به ذکر است پیاده سازی som نیز پیگیری شد که با توجه به حجم بالای داده ها و ابعاد بسیار زیاد آن استفاده از این روش در محیط colab بسیار زمان بر بوده و نیاز به RAM بالایی برای تعداد داده های محدود دارد که عملا تحلیل با تعداد نمونه های ۱۰۰ یا ۲۰۰ تا دقیق نخواهد بود.

همچنین روش دیگری برای صحت سنجی نتایج تمرین به این صورت انجام شده که به کمک روش trunked SVD ابعاد داده‌ها را ابتدا به ۱۵۰ و سپس به کمک ivis به دو بعد کاهش دهیم که در شکل زیر به نمایش داده شده است. همان طور که مشاهده می‌کنید تحلیل های بیان شده در بخش های بالا در شکل زیر نیز صادق است.



شکل ۱۵: ادغام دو روش SVD و IVIS برای کاهش بعد و نمایش نقشه ویژگی حاصل شده