

گزارش پروژه پنجم

اشکان زرخواه ۶۱۰۳۹۹۱۹۶

۱۰ مرداد ۱۴۰۳

نکات اولیه:

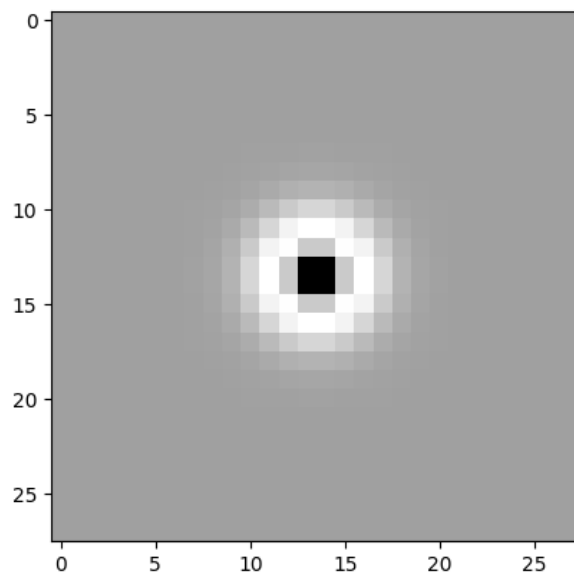
- تمامی نمودارها و تصاویر را می‌توان در پوشه Plots با کیفیت بسیار بالاتر مشاهده کرد.
- تمامی کدها در کنار این گزارش ارائه شده‌اند.
- تمامی مدل‌های نورونی با اختلاف پتانسیل رندوم شروع به شبیه‌سازی می‌کنند.
- تمامی ارتباطات با مقدار اولیه رندوم شروع به شبیه‌سازی می‌کنند.
- تمامی آزمایشات به علت محدودیت محاسباتی و پیچیدگی شبکه عصبی به مدت ۱۰۰ ثانیه با کوچکترین واحد زمانی ۱ ثانیه شبیه‌سازی شده‌اند.
- در آزمایشات مختلف ساختار شبکه ثابت است و بر روی پارامترها و قابلیت‌های ارتباطات و نورون‌ها تمرکز شده است.
- تمامی پارامترها را می‌توان در ابتدا کدهای ضمیمه شده به گزارش پیدا کرد.
- برای هر دو بخش پروژه از مجموعه داده MNIST استفاده شده است، و دلیل این انتخاب ابعاد و تعداد مناسب عکس‌ها بوده است.
- در پیاده‌سازی مدل‌های نورونی تنها از مدل LIF استفاده شده است و از فرمول روبرو برای پیاده‌سازی آن استفاده شده است:

$$\tau_m \frac{du}{dt} = -[u(t) - u_{\text{rest}}] + RI(t)$$

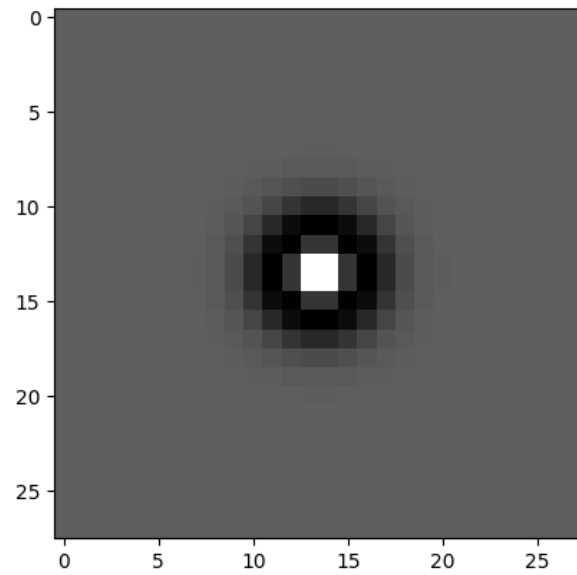
بخش اول:

فیلتر DoG:

فیلتر DoG یا Difference of Gaussians یک تکنیک در پردازش تصویر است که برای شناسایی لبه‌ها و جزئیات در تصاویر به کار می‌رود. این فیلتر بر پایه تفاوت بین دو فیلتر گاوسی با انحراف معیارهای مختلف ساخته شده است. حال در صورتی که انحراف معیار اول از دومی بزرگ تر باشد، مانند شکل ۱، این فیلتر مرکز تیره شده و در غیر این صورت مرکز روشن میشود. و هرچه مقادیر انحراف معیار بزرگتر شود، مانند شکل ۳ حلقه‌ها بزرگتر میشوند و دقت لبه یابی کاهش میابد.

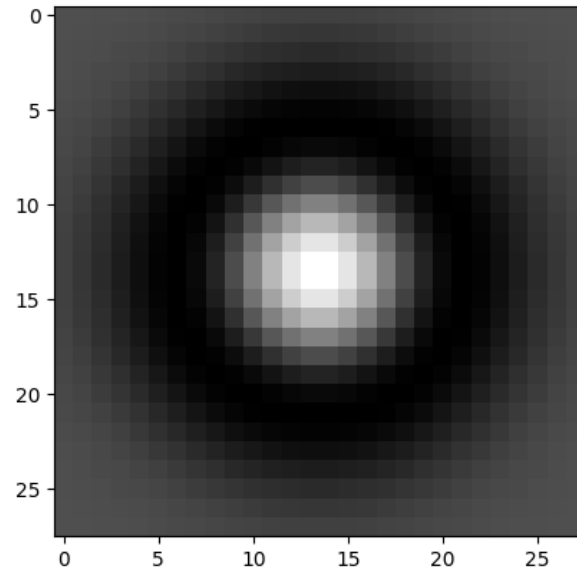


شکل ۱: فیلتر DoG با مرکز تیره



شکل ۲: فیلتر DoG با مرکز روشن

حال تاثیر این فیلتر را میتوان بر روی تصاویر ورودی خام در شکل های ۴ و ۵ به ترتیب با فیلترهای مرکز تیره و مرکز روشن دید.



شکل ۳: فیلتر DoG با مرکز تیره و انحراف معیار بزرگتر.



شکل ۴: اعمال فیلتر DoG با مرکز تیره بر روی تصاویر خام



شکل ۵: اعمال فیلتر DoG با مرکز روشن بر روی تصاویر خام

حال تاثیر این فیلتر را به همراه کدگذاری Time-to-First-Spike میتوان بر روی تصاویر ورودی خام در شکل های ۶ و ۷ به ترتیب با فیلترهای مرکز تیره و مرکز روشن دید.

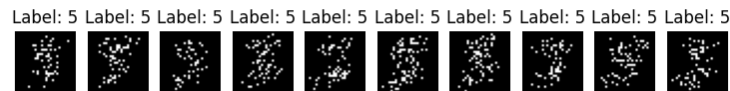
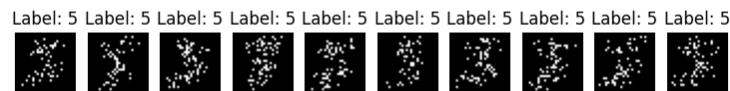


شکل ۶: اعمال فیلتر DoG با مرکز تیره به همراه کدگذاری Time-to-First-Spike بر روی تصاویر خام

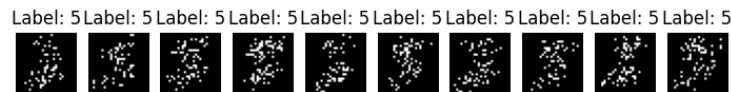
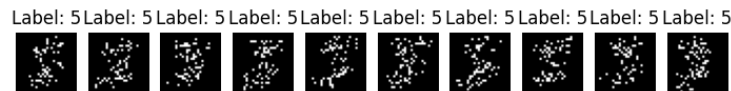


شکل ۷: اعمال فیلتر DoG با مرکز روشن به همراه کدگذاری Time-to-First-Spike بر روی تصاویر خام

حال میتوان تاثیر این فیلتر را به همراه استفاده از توزیع پواسون که در آن هرچه یک پیکسل روشن تر باشد، تعداد ضربه هایش بیشتر است را بر روی یکی از تصاویر ورودی خام در شکل های ۸ و ۹ به ترتیب با فیلتر های مرکز تیره و مرکز روشن دید.



شکل ۸: اعمال فیلتر DoG با مرکز تیره به همراه توزیع پواسون بر روی تصویر خام

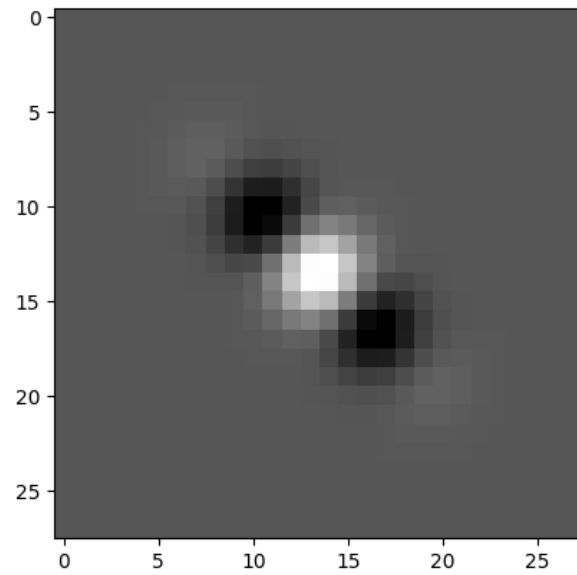


شکل ۹: اعمال فیلتر DoG با مرکز روشن به همراه توزیع پواسون بر روی تصویر خام

فیلتر Gabor:

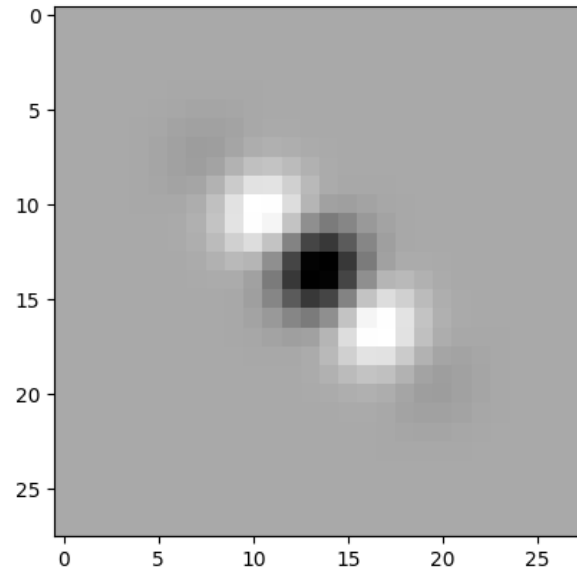
فیلتر Gabor یک ابزار پردازش تصویر است که برای استخراج ویژگی‌های متنی و فرکانسی از تصویر استفاده می‌شود. این فیلتر ترکیبی از یک تابع گاوسی و یک تابع موج‌دار سینوسی است که به تشخیص لبه‌ها، الگوها و ساختارهای محلی کمک می‌کند. فیلتر گابور با تنظیم پارامترهای مختلف قادر به تحلیل ویژگی‌های مقیاس و جهت‌های مختلف است. پارامترهای کلیدی در این فیلتر عبارتند از:

- λ : طول موج‌های بزرگ‌تر باعث تشخیص ویژگی‌های با مقیاس بزرگ‌تر می‌شود، در حالی که طول موج‌های کوچک‌تر برای ویژگی‌های ریزتر مناسب هستند.
 - θ : تغییر این پارامتر باعث می‌شود فیلتر توانایی تشخیص ویژگی‌ها در جهت‌های مختلف را داشته باشد.
 - γ : به تنظیم نحوه پراکندگی سیگنال در جهت‌های مختلف کمک می‌کند. این پارامتر به فیلتر امکان می‌دهد ویژگی‌ها را در الگوهای خاص با نسبت‌های مختلف شناسایی کند.
- حال در شکل ۱۰ و ۱۱ میتوان فیلتر های Gabor با مرکز روشن و تیره و با زاویه ۴۵ درجه را مشاهده کرد.

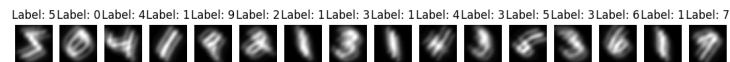


شکل ۱۰: فیلتر Gabor با مرکز روشن

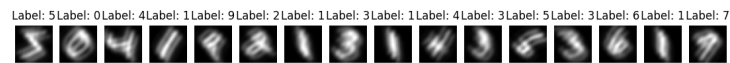
حال تاثیر این فیلتر را میتوان بر روی تصاویر ورودی خام در شکل های ۱۲ و ۱۳ به ترتیب با فیلتر های مرکز تیره و مرکز روشن دید.



شکل ۱۱: فیلتر Gabor با مرکز تیره



شکل ۱۲: اعمال فیلتر Gabor با مرکز تیره بر روی تصاویر خام



شکل ۱۳: اعمال فیلتر Gabor با مرکز روشن بر روی تصاویر خام

حال تاثیر این فیلتر را به همراه کدگذاری Time-to-First-Spike میتوان بر روی تصاویر ورودی خام در شکل های ۱۴ و ۱۵ به ترتیب با فیلتر های مرکز تیره و مرکز روشن دید.

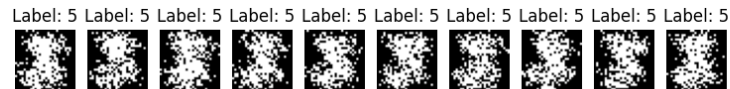


شکل ۱۴: اعمال فیلتر Gabor با مرکز تیره به همراه کدگذاری Time-to-First-Spike بر روی تصاویر خام



شکل ۱۵: اعمال فیلتر Gabor با مرکز روشن به همراه کدگذاری Time-to-First-Spike بر روی تصاویر خام

همانطور که میتوان دید تاثیر هر دو حالت مرکز روشن و مرکز تیره یکسان است و در این فیلتر باید بر روی باقی پارامتر ها برای استخراج بهتر داده تمرکز کرد. حال میتوان تاثیر این فیلتر را به همراه استفاده از توضیح پواسون که در آن هرچه یک پیکسل روشن تر باشد، تعداد ضربه هایش بیشتر است را بر روی یکی از تصاویر ورودی خام در شکل های ۱۶ و ۱۷ به ترتیب با فیلتر های مرکز تیره و مرکز روشن دید.



شکل ۱۶: اعمال فیلتر Gabor با مرکز تیره به همراه توزیع پواسون بر روی تصویر خام

Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5



Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5



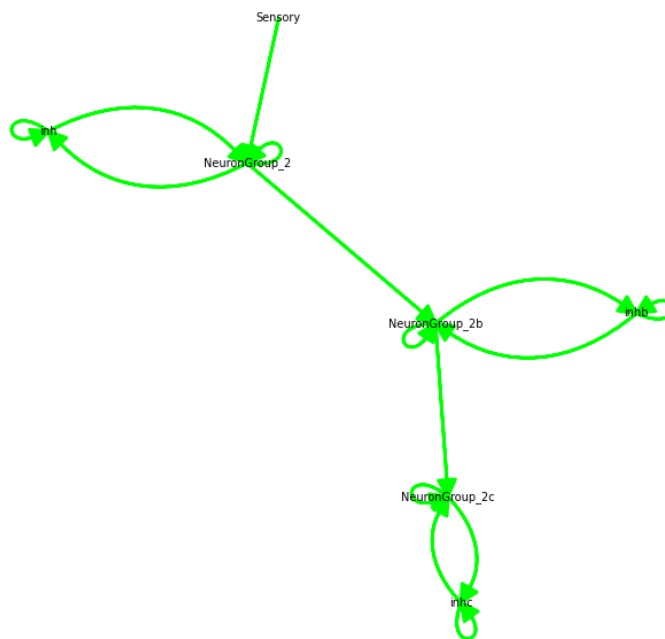
Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5Label: 5



شکل ۱۷: اعمال فیلتر Gabor با مرکز روشن به همراه توزیع پواسون بر روی تصویر خام

بخش دوم:

در این بخش یک شبکه عصبی ضربه ای ساخته شده از ۴ لایه میسازیم که لایه اول آن لایه تصاویر ورودی است که فیلتر DoG با مرکز روشن و توزیع پواسون بر روی آن ها اجرا شده است و ۳ لایه بعدی آن، هر یک از یک جمعیت نورونی excitatory و یک جمعیت نورونی inhibitory تشکیل شده است و هرچه به لایه های انتهایی نزدیک تر میشویم، ابعاد جمعیت ها کاهش میابد. بطور کل ساختار شبکه عصبی را میتوان به صورت شکل ۱۸ نشان داد.

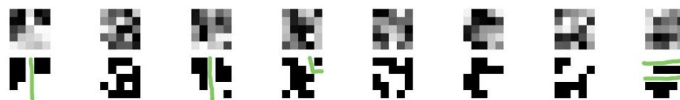


شکل ۱۸: ساختار شبکه عصبی ضربه ای مورد استفاده.

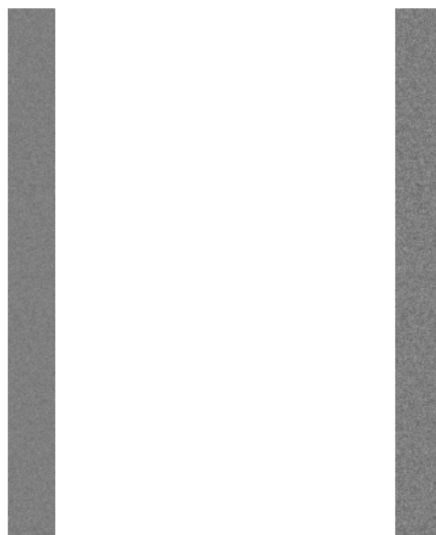
در لایه ورودی، از عمق ۸ استفاده میکنیم تا بتوانیم ۸ ویژگی متفاوت را از ورودی استخراج کنیم و قدرت یادگیری را بالاتر ببریم. ابتدا بدون استفاده از ساز و کار خاصی، تنها این لایه را با Convolution و یادگیری متناسب به لایه بعدی وصل میکنیم و ویژگی استخراج شده در هر لایه را میتوان در شکل های ۱۹ تا ۲۱ دید. هر یک از ویژگی ها به دو صورت مستقیم و فیلتر شده بر حسب مقدار میانگین نمایش داده شده اند.

حال در شکل ۱۹ میتوان دید که شبکه عصبی یادگرفته شده، با آنکه زمان زیادی نداشته است و از ساز و کار خاصی در یادگیریش استفاده نشده است، توانسته است بطور تقریبی خطوط افقی و عمودی را یاد بگیرد، که با خط سبز نمایش داده شده است.

اما از طرفی دیگر میتوان دید که ویژگی های استخراج شده میان لایه های بعدی در شکل های ۲۰ و ۲۱، از پیچیدگی بالاتری برخوردارند و قابل بررسی دقیق نیستند.

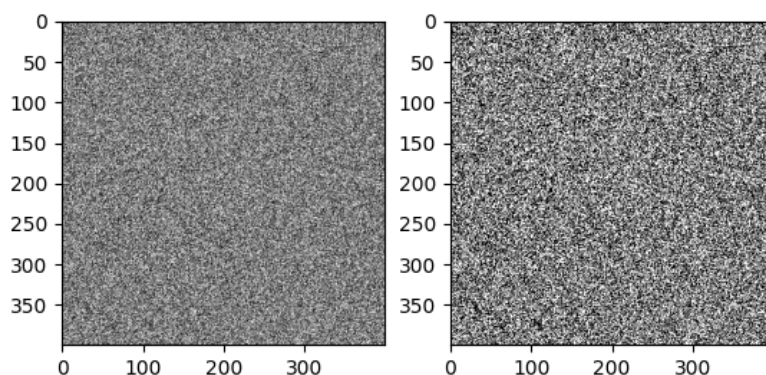


شکل ۱۹: ویژگی های استخراج شده در میان دو لایه اول.



شکل ۲۰: ویژگی های استخراج شده در میان لایه دوم و سوم.

حال میایم و ساز و کار Lateral Inhibition را به لایه ورودی اضافه میکنیم و نتایج جدید را در شکل های ۲۲ تا ۲۴ میتوان دید.

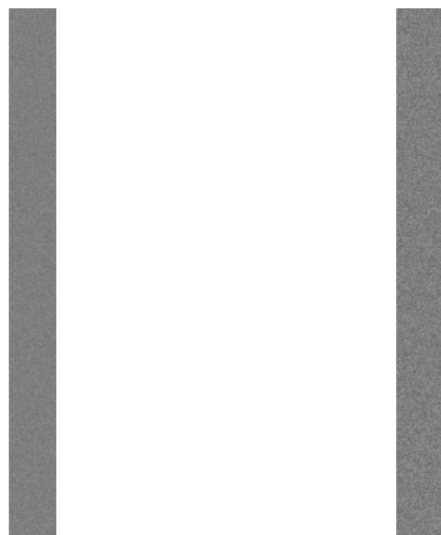


شکل ۲۱: ویژگی های استخراج شده در میان لایه سوم و چهارم.

همانطور که در شکل ۲۲ میتوان دید، قدرت یادگیری افزایش یافته است و میتوان دید که علاوه بر خطوط عمودی و افقی، گوشه ها نیز توسط شبکه یادگرفته شده اند. اما دوباره در مورد ویژگی های یادگرفته شده میان ۳ لایه دیگر نمیتوان به راحتی نظر داد و تنها میتوان گفت که، بنظر میاید (در تصاویر کنار گزارش بهتر دیده میشود) که در میان لایه دوم و سوم، میزان فعالیت بمقداری افزایش یافته است.

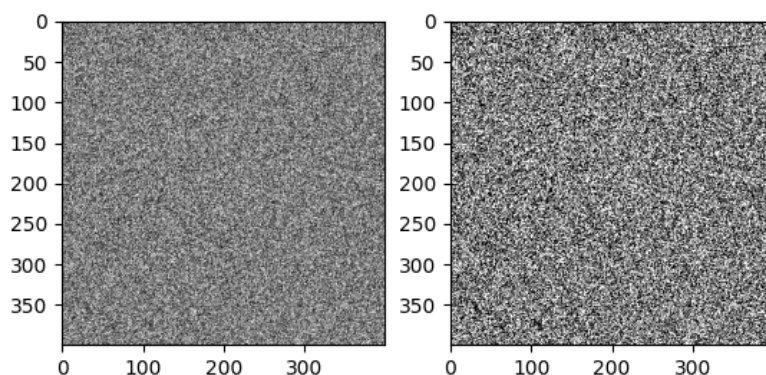


شکل ۲۲: ویژگی های استخراج شده در میان دو لایه اول.



شکل ۲۳: ویژگی های استخراج شده در میان لایه دوم و سوم.

و در انتها میاییم و سازکار kWinnersTakeAll را به نوروں های جمعیت های excitatory اضافه میکنیم. نتایج بدست آمده را میتوان در شکل های ۲۵ تا ۲۷ مشاهده کرد.

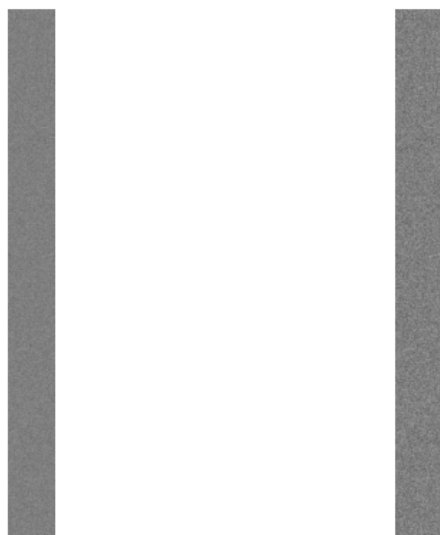


شکل ۲۴: ویژگی های استخراج شده در میان لایه سوم و چهارم.

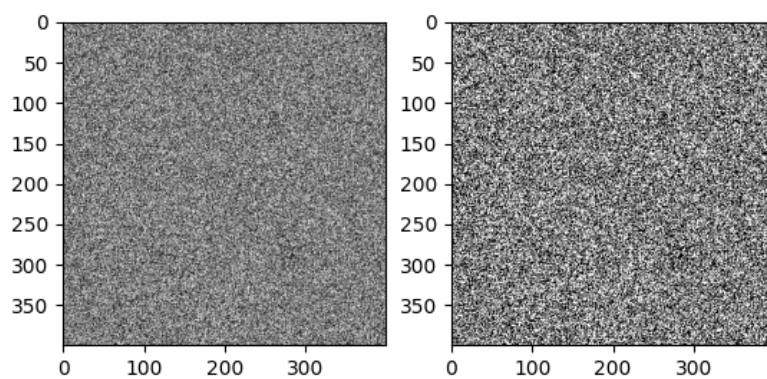
همانطور که انتظار میرفت، میتوان در تمامی شکل ها کاهش فعالیت در شبکه را مشاهده کرد. اما، این کاهش فعالیت موجب یادگیری قطعی تر و دقیق تر شبکه عصبی نشده است. در شکل ۲۵ میتوان دید که الگوی یادگرفته شده در گذشته، در این روش دقت کمتری دارند، و در شکل های ۲۶ و ۲۷ تنها میتوان کاهش فعالیت در شبکه را مشاهده کرد. امکان آن وجود دارد که این روش با مدت زمان بیشتر به یادگیری بهتری برسد، اما میتوان گفت که برای یادگیری با محدودیت محاسباتی خیلی مناسب نمیشد و بر روی عملکرد شبکه تاثیر معکوس دارد.



شکل ۲۵: ویژگی های استخراج شده در میان دو لایه اول.



شکل ۲۶: ویژگی های استخراج شده در میان لایه دوم و سوم.



شکل ۲۷: ویژگی های استخراج شده در میان لایه سوم و چهارم.