

## تمرین سری چهارم شبکه عصبی-سارا حسینی فشتامی

### سوال ۱-

۱- undercomplete: این اتوانکدرها تعداد نورون های کمتری در لایه هیدن خود در مقایسه با لایه ورودی دارند. این شبکه را وادار می کند تا یک نمایش فشرده از داده های ورودی را بیاموزد. اتوانکدرهای ناقص معمولاً برای کاهش ابعاد، استخراج ویژگی و حذف نویز داده ها استفاده می شوند. مثلاً یک اتوانکدر با یک لایه ورودی به اندازه 100 و یک لایه پنهان به اندازه 50 یک اتوانکدر ناقص است، زیرا لایه هیدن دارای نورون های کمتری است. این می تواند هنگام کار با داده های با ابعاد بالا که هدف آن کاهش نویز و گرفتن مهمترین ویژگی ها است مفید باشد.

۲- overcomplete: این اتوانکدرها تعداد نورون های بیشتری در لایه هیدن در مقایسه با لایه ورودی دارند. اتوانکدرهای اورکامپلیت می توانند با کشف الگوها در داده های ورودی، نمایش های مفیدی را بیاموزند. مثال: یک اتوانکدر با یک لایه ورودی به اندازه 100 و یک لایه پنهان به اندازه 200 یک اتوانکدر اورکامپلیت است. ممکن است در anomaly detection استفاده شود،

۳- exactly complete: این اتوانکدرها به اندازه لایه ورودی دارای تعداد نورون در لایه هیدن هستند. اتوانکدرهای کاملاً کامل هم می توانند نمایش های فشرده داده های ورودی را با داشتن توابع فعال سازی غیر خطی بیاموزند و همچنین می توانند در سناریوهایی استفاده شوند که حفظ ابعاد داده ها مهم است. مثال: یک اتوانکدر با لایه های ورودی و هیدن با اندازه 100 یک AutoEncoder کاملاً کامل است. این می تواند هنگام برخورد با مجموعه داده هایی که ساختار ذاتیشان را میخواهیم حفظ کنیم بکار بیاید.

### سوال ۲-

اتوانکدرها هدفشان بازسازی ورودی توسط یادگیری یک لیتنت اسپیس است و به حداقل رساندن تفاوت بین ورودی و خروجی بازسازی شده. از آنجایی که هدف آنها کشف ساختار درون داده ها است، اتوانکدرها را یادگیری بدون نظارت میدانیم. یعنی فقط  $X$  داریم و  $y$  یعنی لیبیل در کار نیست و نبودن این لیبیل ها اتوانکدرها را مجبور می کند که نمایشی معنادار از داده بیاموزند.

مدل‌های یادگیری با نظارت: در مقابل، مدل‌های یادگیری با نظارت برای یادگیری ارتباط بین داده‌های ورودی و لیبل‌ها آموزش داده می‌شوند.

از اتوانکدرها می‌توان برای pretrain کردن و یادگیری یک سری روابط و ساختارهای اولیه قبل از فاین‌تیون کردن با داده‌های برچسب‌دار استفاده کرد. این کار باعث می‌شود عملکرد یادگیری تحت نظارت بعدی هنگامی که داده‌های برچسب‌دار کمیاب هستند بهتر شود.

از اتوانکدرها می‌توان برای کاهش ابعاد و فیچر اکسترکشن زمانی که داده‌های ورودی دارای ابعاد بالا یا نویز هستند استفاده کرد. این می‌تواند با کاهش پیچیدگی داده‌ها به بهبود عملکرد مدل‌های یادگیری با نظارت کمک کند.

مدل‌های جنریتیو: از اتوانکدرها می‌توان برای ساخت مدل‌های مولد مانند (VAE) و (GAN) استفاده کرد. به طور کلی، در حالی که اتوانکدرها و مدل‌های یادگیری نظارت شده کلاسیک اهداف متفاوتی دارند، می‌توان آن‌ها را به روش‌های مختلفی ترکیب کرد تا انواع مختلفی از مشکلات یادگیری ماشین را برطرف کند.

### سوال ۳-

اتوانکدرها با به حداقل رساندن خطای بازسازی، قصد دارند نمایش خوبی از داده‌های ورودی را بیاموزند. برای تعیین کمیت این خطا می‌توان از توابع مختلف loss استفاده کرد و انتخاب بستگی به ماهیت داده‌ها و تسک ما دارد.

(MSE) یک تابع هزینه محبوب برای مسائل رگرسیون است. MSE برای داده‌های با مقدارهای عدد حقیقی مناسب است و مشتق پذیر است و همین است که بهینه‌سازی آن را آسان می‌کند، اما به اوتلایرها حساس است. مثلاً برای داده‌های پیوسته، مانند مقادیر پیکسل‌ها در یک تصویر، MSE اغلب برای ارزیابی کیفیت بازسازی استفاده می‌شود.

(BCE): معمولاً برای تسک‌های کلاس‌بندی باینری استفاده می‌شود. احتمال پیش‌بینی کلاس صحیح را برای ورودی و خروجی بازسازی شده اندازه می‌گیرد و از این روش، میزان تشابه بین ورودی و خروجی را می‌سنجد. BCE برای داده‌های باینری مناسب است و همچنین مشتق پذیر است.

مثلا اگر اتوانکدر را برای تسک های کلاسیفیکیشن باینری بخواهیم یا داده ها ویژگی های باینری داشتند، BCE مناسب است

(KLD): که به عنوان آنتروپی نسبی نیز شناخته می شود، تفاوت بین دو توزیع احتمال را اندازه گیری می کند. در زمینه اتوانکدرها، از KLD می توان برای به حداقل رساندن تفاوت بین توزیع داده های ورودی و خروجی بازسازی شده استفاده کرد. KLD معمولاً در (VAE) ها استفاده می شود تا اطمینان حاصل شود که فضای لیتنت آموخته شده از توزیع دلخواه پیروی می کند. در VAE ها، KLD به یادگیری یک فضای لیتنت فشرده و پیوسته کمک می کند، که امکان تولید نمونه های جدید را هم بهبود می بخشد.

به طور خلاصه، MSE برای داده های پیوسته و با مقادیر حقیقی مناسب است، BCE برای داده های باینری مناسب است، و KLD معمولاً در VAE ها برای یادگیری یک فضای لیتنت مناسب استفاده می شود.

#### سوال ۴-

از طرق زیر میتوان ارزیابی کرد:

-خطای بازسازی: یعنی تفاوت بین ورودی اصلی و خروجی بازسازی شده. مثل MSE.

-توسط چشم ارزیابی کردن

-کاهش ابعاد: از اتوانکدرها می توان برای کاهش ابعاد با فشرده سازی ورودی در فضای لیتنت استفاده کرد. برای

ارزیابی این فرآیند، می توانیم با استفاده از تکنیک هایی مانند PCA یا t-SNE فضای لیتنت را ویژوالایز کنیم. اگر داده ها به خوبی از هم جدا شده باشند، نشان می دهد که اتوانکدر ما ویژگی های مفیدی را یاد گرفته است.

-الگوریتم های کلاسترینگ (مثلاً K-means) را روی لیتنت اسپیس اعمال کنیم. اگر نتایج کلاسترینگ با ساختار داده ها یکی باشد، نشان می دهد که اتوانکدر ویژگی های خوبی را یاد گرفته.

-استفاده در ترنسفر لرنینگ: می توانیم از لیتنت اتوانکدر به عنوان فیچرهای ورودی برای یک تسک دیگر استفاده کنیم و ببینیم چقدر عملکرد را بهبود میدهد.

-تحلیل روبااست بودن: توانایی اتوانکدر را برای می توانیم برای بازسازی ورودی تحت تغییر شکل های مختلف

(مانند نویز، روتیشن یا اسکیل کردن) بسنجیم. اگر اتوانکدر همچنان بتواند تصویر اصلی را بازسازی کند، نشان میدهد که مدل قوی است.

## سوال ۵-

ترفند reparameterization تکنیکی است که در (VAEs) استفاده می‌شود تا بکپروپ گرادیان‌ها را در حین فرآیند سمپلینگ هم فعال کند. در VAE ها، بخش انکدر مجموعه‌ای از پارامترها (میانگین و واریانس) را خروجی می‌دهد که توزیع احتمال لیتنت را توصیف می‌کند که از آن یک سمپل نمونه برداری می‌شود. با این حال، فرآیند سمپلینگ یک طبیعت تصادفی یا استوکستیک دارد، که در محاسبه گرادیان‌ها چالش ایجاد میکند. ترفند reparameterization یعنی مؤلفه‌های قطعی و تصادفی را در فرآیند سمپلینگ جدا کنیم تا مشکل بالا حل شود.

چون عملاً ما در حین فرآیند سمپل، داریم:

$$z \sim \mathcal{N}(\mu, \sigma)$$

و نمیتوانیم به شبکه بگوییم که در سمپل گیری، ارور داشته‌ای و دفعه بعد جوری سمپل بگیر که خطا کم شود. فرآیند تصادفی در اینجا با میانگین و واریانس آن توزیع لیتنت، در هم تنیده شده اند. ما با reparameterization ، این درهم‌تنیدگی را باز میکنیم تا بتوانیم ارور را به میانگین و واریانس برگردانیم.

$$z \sim \mu + \sigma \odot \varepsilon$$

$$\varepsilon \sim \mathcal{N}(0, 1)$$

## سوال ۶-

-beta VAE: عبارت KL در تابع loss را ضربدر یک بتا میکنیم. (خود KL هم یک رگولایزر است که باعث میشود توزیع لیتنت گاسی یا برنولی یا ... باشد) افزایش بتا VAE را تشویق می‌کند تا یک نمایش لیتنت گسسته‌تر و معنادارتر را بیاموزد.

آموزش Adversarial : از آموزش Adversarial نیز میتوان برای گسستگی لیتنت استفاده کرد. مثلاً، در روشی به نام Adversarial-VAE، یک شبکه Adversarial اضافی آموزش میبند که کد لیتنت را بر اساس تصویر تولید شده پیشبینی کند. سپس انکدر برای فریب دادن این شبکه آموزش داده می‌شود که منجر به نمایشی از هم گسیخته‌تر و قوی‌تر می‌شود.

independent Component Analysis: ادغام (ICA) در VAE ها هم می تواند مفید باشد. هدف روش های مبتنی بر ICA جداسازی یک سیگنال چند متغیره به اجزای فرعی است که مستقل از یکدیگر هستند.

## سوال ۷-

مشکلاتی که این داده ها ایجاد میکنند:

ورودی های با ابعاد بالا می توانند یادگیری لیتنت را برای VAE دشوار کنند. مدل برای به دست آوردن همبستگی ها و الگوهای پیچیده موجود در داده ها به چالش می خورد. توزیع های پیچیده داده با الگوهای غیرخطی می تواند برای VAE ها چالش برانگیز باشد. VAE های استاندارد اغلب یک توزیع گاوسی ساده را برای هم prior و هم posterior فرض می کنند که ممکن است برای داده های پیچیده مناسب نباشد. همچنین، VAE ها ممکن است به مشکل posterior collapse بخورند، که در آن مدل داده های ورودی را نادیده می گیرد و فقط یاد می گیرد که بر اساس توزیع prior، خروجی تولید کند.

روش های حل این مشکلات:

- معماری عمیق تر

- استفاده از prior های پیشرفته تر بجای گاوسی

- Wasserstein VAEs عوض کردن تابع هدف VAE با فاصله Wasserstein بین توزیع مدل و توزیع داده واقعی.

- بچ نورمالیزیشن هم باعث استیبل شدن آموزش، بهبود همگرایی و کاهش مشکل posterior collapse میشود.

- استفاده از دراپ اوت یا رگولاریزیشن های ال نورم.

- استفاده از بهینه ساز ADAM یا هر گونه learning rule که adaptive باشد.

## سوال ۸-

- conditional vae: در یک VAE شرطی (CVAE)، مدل یاد می گیرد که نمونه هایی را مشروط به اطلاعات اضافی، مانند لیبل کلاس ها یا سایر ورودی های کمکی تولید کند. این با معرفی یک ورودی اضافی به شبکه انکدر

و دیگر که اطلاعات شرط را نشان می دهد، به دست می آید. CVAE ها باعث تولید داده های متنوع میشوند، که میتواند در تسکهای مانند تولید تصویر، تولید متن و تشخیص ناهنجاری مفید باشد.

یادگیری نیمه نظارت شده: VAE ها را می توان برای یادگیری نیمه نظارتی با گنجاندن اطلاعات لیبِل در مدل استفاده کرد. میتوانیم یک loss از کلسیفیکیشن هم به تابع هدف اضافه کنیم و فضای لیتنت را لرن کنیم که طبق لیبِل های کلاسها ساختار یابد. این کار به VAE ها اجازه می دهد تا از داده های برچسب دار و بدون برچسب استفاده کنند و عملکرد مدل را در سناریوهایی که داده های برچسب دار کمیاب هستند، بهبود می بخشد.

لیتنت از هم گسیخته: VAE ها را می توان لرن کرد که نمایش هایی از هم گسیخته در فضای لیتنت بدست بیاورد که در آن هر بعدی ویژگی های متمایز و قابل تفسیر را دربردارد. این را می توان با معرفی رگولاریزیشن ها یا محدودیت هایی که استقلال را در بین ابعاد فضای لیتنت بوجود بیاورند، به دست آورد. نمایش های تفکیک شده باعث تفسیرپذیری بهتر، تولید کنترل شده ی داده ها و قابلیت جنرالیزیشن بهتر میشوند.

- VAE های سلسله مراتبی (HVAEs) از یک فضای لیتنت چندلایه برای ثبت روابط و ساختارهای سلسله مراتبی در داده ها استفاده می کنند. هر لایه از فضای لیتنت مربوط به سطح مختلفی از انتزاع است، که به مدل اجازه می دهد تا توزیع داده های پیچیده را بهتر یاد بگیرد. HVAE ها قابلیت تولید داده بهتری دارند.

## سوال ۹-

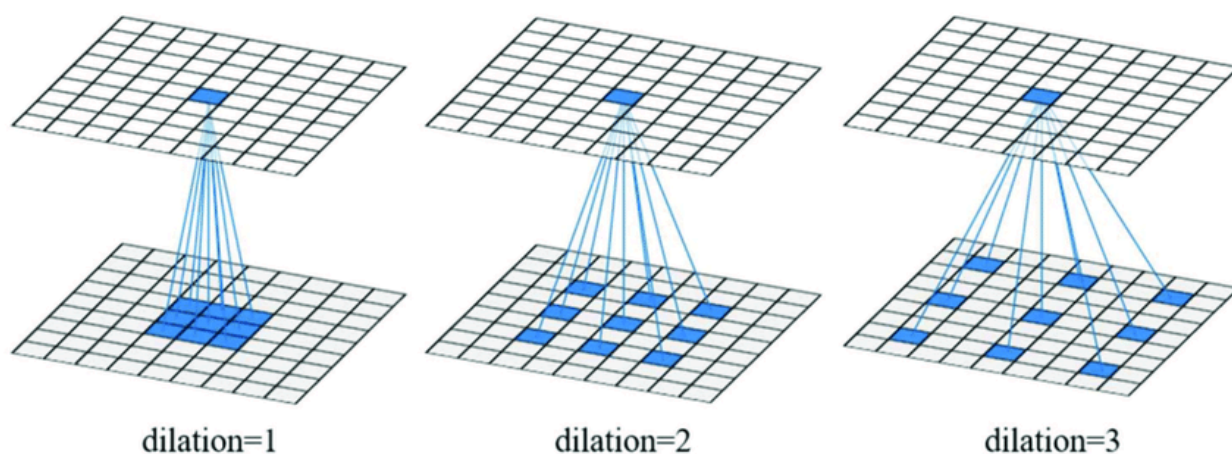
۱- لایه کانولوشن استاندارد: این ابتدایی ترین نوع لایه کانولوشن است که در آن مجموعه ای از فیلترهای learnable روی داده های ورودی پیچش میکنند  $x \times d$  تا یک یا چندین فیچر مپ دو بعدی تولید کنند. از آنها در CNN ها برای image classification, object detection, semantic segmentation و بسیاری کاربردهای دیگر استفاده میکنیم.

۲- Depthwise Separable Convolutional Layer: این لایه کانولوشن های عمقی انجام می دهد، یعنی هر چنل ورودی با یک فیلتر مجزا کانوالو می شود. اغلب بعد آن نیز یک کانولوشن نقطه ای داریم که خروجی های

کانولوشن عمقی را ترکیب کند. این کانولوشنهای عمقی، پیچیدگی محاسباتی را کاهش می‌دهند و مناسب برنامه‌های موبایل و embedded هستند که منابع محدودی دارند.

۳. کانولوشن ترنسپوز: همان دیکانولوشن، عمل معکوس یک کانولوشن استاندارد را انجام می‌دهد. یاد می‌گیرد که داده‌های ورودی را با افزایش ابعاد مکانی آن upsample کند. دیکانولوشن‌ها معمولاً در مدل‌های مولد مانند (VAE) و (GAN) و همچنین در image segmentation کاربرد دارند.

۴. کانولوشن دایلیتد: این لایه از فیلترهایی بصورت زیر استفاده می‌کند که بدون افزایش تعداد پارامترها یا هزینه محاسباتی، میدان دید بزرگتری را امکان پذیر می‌کند. این کانولوشن‌ها در کاربردهایی مانند semantic segmentation مفید هستند و عمل نقشاتنشن را بازی میکنند و روابط دورادور را محاسبه میکنند.



همچنین انواع دیگر ConvLSTM و Graph Convolutional Layer هستند.

سوال ۱۰-

توضیح در سوال قبل.

کاربردهای بیشتر:

در شبکه موبایل نت برای کاهش محاسبات.

در شبکه VAN برای درآوردن اتنشن عکس و long range dependence ها.

در سمنتیک سگمنتیشن چون میدان دید بزرگ اینجا مهم است تا یکپارچگی تصویر را ببینیم.

در پردازش صدا نیز مفید است مثل speech recognition, music classification, speaker identification.