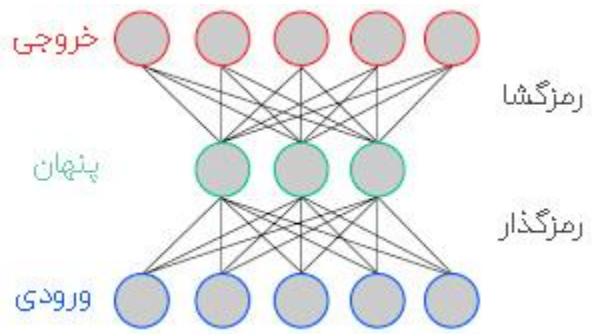
Exercise 1

How does the VAE architecture allow it to generate new data points, especially compared to associative auto-encoder, which cannot generate new data points?

Autoencoder یک نوع خود رمزگذار است که یک تصویر را به عنوان ورودی دریافت و با استفاده از تعداد بیت محدود تر ان را دوباره میسازد.
Autoencoder از یک بخش دیکودر و یک بخش اینکودر و یکی سری لایه پنهان میان اینها تشکیل شده است.
ایده کلی ر بسیار ساده است و شامل تنظیم یک رمزگذار و رمزگشا به عنوان شبکه های عصبی و یادگیری بهترین طرح رمزگذاری-رمزگشایی با استفاده از یک فرآیند بهینه سازی تکراری است. بنابراین، در هر تکرار، معمار مدل را با مقداری داده تغذیه می‌کنیم، خروجی رمزگذاری‌شده-رمزگشایی‌شده را با داده‌های اولیه مقایسه می‌کنیم و خطا را در معماری منتشر می‌کنیم تا وزن‌های شبکه‌ها به‌روزرسانی شود.



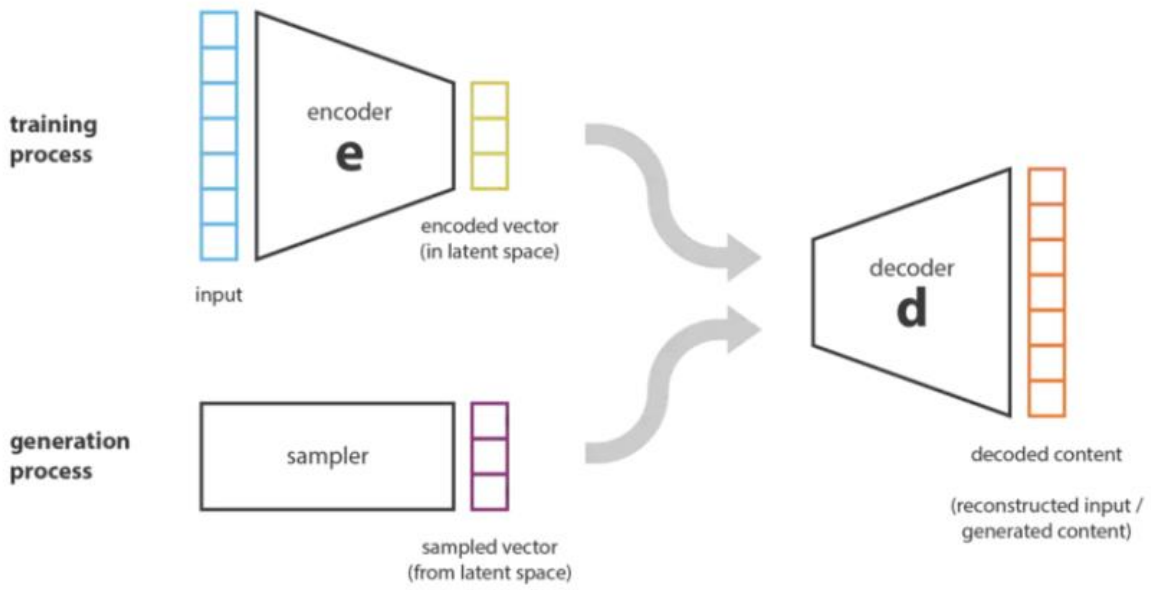
انواع مختلفی از Autoencoder ها وجود دارند مثل Variational Autoencoder و Sparse Autoencoder و مدل Denoising Autoencoder که موضوع مورد بحث ما مورد اول میباشد.

رمزگذار ورودی را فشرده می کند و رمزگشا سعی می کند ورودی را از نسخه فشرده ارائه شده توسط رمزگذار بازآفرینی کند.
Autoencoder یک شبکه عصبی است که برای کپی کردن ورودی خود در خروجی آموزش دیده است. رابطه رمزگذارهای خودکار با تولید محتوا کمی پیچیده تر از حدس های ساده است. افزودن توابع غیر خطی (مانند توابع فعال سازی غیرخطی و لایه های پنهان بیشتر) باعث می شود Autoencoder بتواند نمایش های نسبتاً قدرتمندی از داده های ورودی در ابعاد پایین تر را با از دست دادن اطلاعات بسیار کمتر یاد بگیرد.

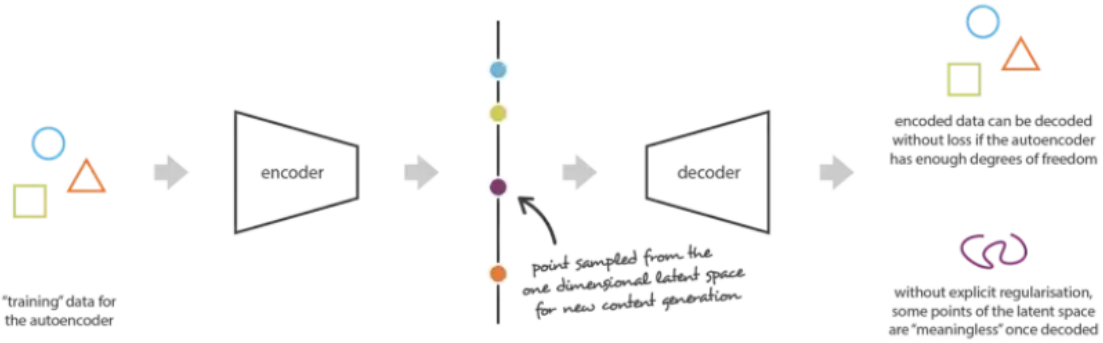
بیایید اکنون لینک مشکل تولید دیتا را بررسی کنیم و محدودیت‌های Autoencoder را به شکل فعلی برای این مشکل ببینیم و در نهایت VAE ها را معرفی کنیم.

از تفاوت های این دو مدل میتوان به این مورد اشاره کرد که رمزگذار در Autoencoder ردارهای پنهان را خروجی می دهد. به جای خروجی بردارها در فضای پنهان، رمزگذار در VAE پارامترهای یک توزیع از پیش تعریف شده را در فضای پنهان برای هر ورودی خروجی می دهد. سپس VAE محدودیتی را بر این توزیع پنهان تحمیل می‌کند و آن را مجبور می‌کند که توزیع نرمال باشد.

پس تفاوت اصلی این است که ورودی به جای یک بردار به دو بردار کدگذاری می شود. این دو بردار برای تعریف یک توزیع نرمال استفاده می‌شوند، جایی که نمایش بنهان ورودی از آن گرفته می‌شود.
در واقع، هنگامی که Autoencoder آموزش داده شد، ما هم یک رمزگذار و هم رمزگشا داریم، اما هنوز راهی واقعی برای تولید محتوای جدید نداریم. می‌توانیم این طور فکر کنیم که اگر فضای پنهان به اندازه کافی منظم باشد (به‌خوبی توسط رمزگذار در طول فرآیند آموزش سازماندهی شده باشد)، می‌توانیم به‌طور تصادفی نقطه‌ای را از آن فضای پنهان برداریم و آن را رمزگشایی کنیم تا یک فضای (دیتا) جدید به دست آوریم. سپس رمزگشا کمابیش مانند مولد یک شبکه خصمانه مولد عمل می کند.



ما یک رمزگذار و یک رمزگشا را توصیف کردیم که به اندازه کافی قدرتمند باشد تا هر تعداد داده آموزشی اولیه را بر روی محور واقعی قرار دهد (هر نقطه داده به عنوان یک مقدار واقعی کدگذاری می‌شود) و آنها را بدون هیچ ضرر بازسازی رمزگشایی می‌کند.
در چنین حالتی، درجه آزادی بالا که امکان رمزگذاری و رمزگشایی را بدون از دست دادن اطلاعات (با وجود ابعاد کم فضای پنهان) موجب میشود، منجر به اورفیتینگ می‌شود که به این معنی است که برخی از نقاط فضای پنهان محتوای بی‌معنی می‌دهند.



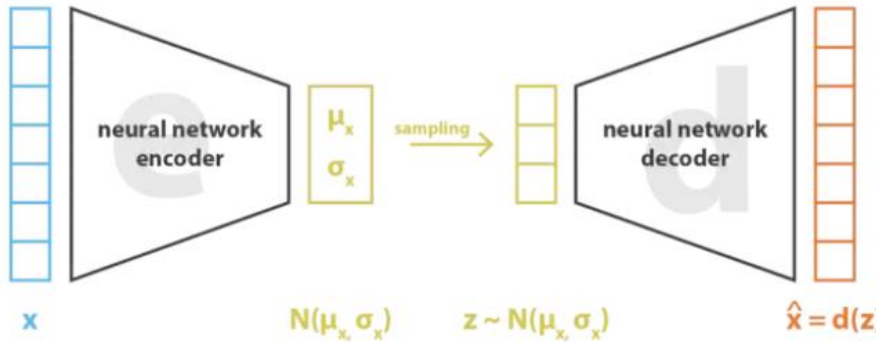
برای اینکه بتوانیم از رمزگشای Autoencoder بر ای اهداف تولیدی استفاده کنیم، باید مطمئن باشیم که فضای پنهان به اندازه کافی منظم است. بنابراین در نهایت Variational Autoencoder را می توان به عنوان یک مدل مناسب تعریف کرد که آموزش آن برای جلوگیری از اورفیتینگ منظم شده است و اطمینان حاصل می کند که فضای پنهان دارای ویژگی های خوبی است که فرآیند تولید دیتا را امکان پذیر می کند.

اختلاف VAE با Autoencoder که در ان با برخلاف Autoencoder ما از فضای پنهان بین اینکودر و دیکودر اطلاع داریم و میدانیم مفادیر در چه رنجی قرار خواهند گرفت و این در تولید دیتای جدید بسیار کمک کننده خواهد بود. VAE از یک توزیع نرمال احتمالی کمک میگیرد تا در فضای پنهان به جای بردار اعداد از توزیع های احتمالی تولید کند. برخلاف مدل های دیگر این مدل در تولید دیتای جدید تناظر نظیر به نظیر هدفش نیست بلکه هدفش رسیدن به قسمتی از لایه پنهان با استفاده از ورودی و رسیدن از همان قسمت به خروجی است.

ابتدا، ورودی به عنوان یک توزیع در فضای پنهان کدگذاری می شود، سپس یک نقطه در فضای پنهان از آن توزیع به عنوان نمونه انتخاب می شود و نقطه نمونه برداری شده رمزگشایی می شود و پس از آن خطای ان محاسبه می شود. در نهایت، خطای ان از طریق شبکه منتشر می شود.

توزیع‌های کدگذاری‌شده به صورت نرمال انتخاب می‌شوند تا رمزگذار آموزش ببیند تا میانگین و ماتریس کوواریانس را که این گاوسی ها را توصیف می‌کند، برگرداند. دلیل اینکه یک ورودی به‌جای یک نقطه منفرد به‌عنوان توزیعی با مقداری واریانس کدگذاری می‌شود، این است که بیان منظم‌سازی فضای پنهان را به‌طور طبیعی امکان‌پذیر می‌کند و توزیع‌های بازگردانده‌شده توسط رمزگذار مجبور می‌شوند نزدیک به یک توزیع نرمال استاندارد باشند.

بنابراین، تابع ضرری که هنگام آموزش یک VAE به حداقل می رسد، از یک "ترم بازسازی" (در لایه نهایی) تشکیل شده است که تمایل دارد تا طرح رمزگذاری-رمزگشایی را تا حد امکان کارآمد کند، و یک "ترم تنظیم" (در مورد لایه پنهان)، که تمایل دارد سازماندهی فضای پنهان را با نزدیک کردن توزیع‌های بازگردانده شده توسط رمزگذار به توزیع نرمال استاندارد، منظم کند.



$$loss = \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

نظمی که فضای پنهان برای فرآیند تولید داده باید داشته باشد را می توان از طریق دو ویژگی اصلی بیان کرد: 1-پیوستگی (دو نقطه نزدیک در فضای نهان پس از رمزگشایی نباید دو محتوای کاملاً متفاوت را ارائه دهند) 2- کامل بودن (برای توزیع انتخابی، یک نقطه نمونه برداری شده از فضای پنهان باید پس از رمزگشایی محتوای معنی داری بدهد)

Exercise 2

Variational auto-encoders optimize a lower bound of the data likelihood for a given input sample x (i) such that

$$\mathcal{L}(\theta, \phi; x^{(i)}) = \mathrm{E}_{q_\phi(z|x^{(i)})}[log\, p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)).$$

Explain the task of the KL−divergence term.

Write down the advantage of modeling (z) and $q\phi(z|x\ (i)\ )$ by using Normal distribution with a diagonal covariance matrix.

Explain the task of the first term and its effect on the latent space.

در سوال قبل دیدیم که تابع ضرری که هنگام آموزش یک VAE به حداقل می رسد از یک "ترم بازسازی" (در لایه نهایی) تشکیل شده است، که تمایل دارد طرح رمزگذاری-رمزگشایی را تا حد ممکن کارآمد کند، و یک "ترم تنظیم" (در لایه پنهان) ، که تمایل دارد سازماندهی فضای پنهان را با نزدیک کردن توزیع‌های بازگردانده شده توسط رمزگذار به توزیع نرمال استاندارد، منظم کند. این عبارت منظم‌سازی به‌عنوان واگرایی کولبک-لایبر(KL) بین توزیع برگشتی و یک گاوسی استاندارد بیان می‌شود و جلوتر بیشتر توجیه خواهد شد. می‌توانیم متوجه شویم که واگرایی کولبک-لایبلر بین دو توزیع گاوسی شکل بسته‌ای دارد که می‌تواند مستقیماً بر حسب میانگین و ماتریس‌های کوواریانس دو توزیع بیان شود. کولبک-لایبر در اصل یک برای تشخیص میزان به هم نزدیک بودن دو توزیع احتمالی است. این ترم در تابع لاس وجود دارد تا با استفاده از ان دو توزیع احتمالی ورودی این تابع را در عمل لرنینگ به هم نزدیک کنیم.

$$D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)).$$

همان طور که دیده میشود دو ورودی دارد.

$q_\phi(z|x^{(i)})$: که همان توریع احتمالی ورودی ایکودینگ ما است

$p_\theta(z)$: یک توزیع احتمالی ثابت که ما در نظر میگیریم تا ورودی اولیه به ان نزدیک شود.

هدف از انجام اینکار افزایش دامنه عمل برای تولید دیتای جدید میباشد. اگر این ترم وجود نداشت دامنه تا حد زیادی به Autoencoder معمولی شبیه میشد.

همانطور که دیدم $p_0(z)$ مقدار ثابتی است که خودمان در نظر میگیریم. پس انتخاب این عدد ثابت مهم خواهد شد زیرا از طرفی پیوسته بودن ان باعث عدم دو نقطه احتمالی در فضای پنهان بعد از رمزگشایی خواهد شد و از طرف دیگر کامل بودن ان موجب معنی دادن توزیع احتمالی انتخابی در فضای نمونه بعد از رمزگشایی خواهد شد. در صورتی که این ثابت این ویژگی هارا داشته باشد یک فضای پنهان منظم و در غیر این دو صورت فضای پنهان نامنظم خواهیم داشت. پس برای اینکه فضای پنهان منظم شود نیاز به منظم کردن هم ماتریس کوواریانس و هم میانگین توزیع های برگردانده شده توسط رمزگشا خواهیم داشت. چون در این عمل $p_0(z)$ یک توضیع نرمال خواهد بود پس ماتریس کوواریانس باید به ماتریس همانی نزدیک شود تا Punctual distribution رخ ندهد.

از طرف دیگر میانگین توزیع های برگردانده شده هم باید به صفر نزدیک باشد تا توزیع های رمزگذاری شده از هم فاصله زیادی نداشته باشند. نتیجه میگیریم وجود این دو مورد بهن طور کلی کمک به رعایت دو مورد اصلی ثابت و بهبود مدل میکند.

حال گر بررسی کنیم که ترم اول چه تاثیری میگذارد به احتمال میرسیم:

$$\mathcal{L}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)).$$

این بخش همان توزیع احتمالی است که از طریق فرایند کدگزاری بدست امده است. به عبارت دیگر یعنی به شرطی که ایکس آی ام را به تابع کیو دهیم یک توزیع احتمالی بدست اورد.

ای همان امید ریاضی است که به این معناست که اگر توزیع ایکس آی ام را به تابع اینکودر دهیم تابع توزیع احتمالی کیو بدست می اید. حال اگر از خود این تابع نمونه ی زد را بگیریم انتظار داریم که به ازای هر زد اگر زد را به دیکودر دهیم داده ای مشابه ایکس آی ام به ما دهد.

ترم اول در اصل همان احتمال رخ داد است به این صورت که میگوید چه ایکس آی هایی رخ میدهند به شرطی که مولفه ی زد انها رعایت شود. به عبارت دیگر یک نقطه از زد را داریم که میخواهیم به ایکس متصل کنیم.)

به عبارت دیگر این ترم نهایتا تلاش میکند وزن های کدگشا و کدگذار به گونه ای قرار داده شوند که در نهایت داده ورودی ایکس آی ام را در خروجی داشته باشیم.