

## سوال ۱:

از آنجا که شبکه های عصبی قادر به یادگیری روابط غیرخطی هستند، یک AE با توابع فعال سازی غیرخطی را میتوان به عنوان یک تعمیم قدرتمندتر (غیرخطی) از الگوریتم تحلیل مولفه های اساسی PCA تصور کرد. در حالیکه PCA سعی در کشف یک ابرصفحه در ابعادی پایین تر دارد که داده های اصلی را توصیف کند، خودرمنگار غیرخطی قادر به یادگیری منیفولدهای غیر خطی هستند (یک منیفولد به صورت ساده به عنوان یک سطح پیوسته و غیر متقاطع تعریف می شود).

PCA نسبت به رمزگذارهای خودکار سریعتر و کم هزینه تر است و کاملاً شبیه یک رمزگذار خودکار تک لایه با عملکرد فعال سازی خطی است. به دلیل تعداد زیاد پارامترها، رمزگذار خودکار مستعد بیش از حد برازش است.

برای اینکه کاهش ابعاد موفقیت آمیز باشد، باید ساختاری با ابعاد پایین در فضای ویژگی وجود داشته باشد. به بیان دیگر، ویژگی ها باید به یکدیگر مرتبط باشند. اگر ساختار ابعاد پایین غیرخطی باشند یا انحنا داشته باشد، رمزگذارهای خودکار ممکن است اطلاعات بیشتری را با ابعاد کمتر رمزگذاری کنند. در نتیجه، در موارد خاص، آنها یک استراتژی کاهش ابعاد برتر نسبت به PCA هستند.

---

## سوال ۲:

۱. کاهش ابعاد (از بین بردن پیچیدگی):

AE برای کاهش پیچیدگی داده های ورودی استفاده میشود. با کاهش تعداد مقادیر ورودی، جزئیات کوچک (و نامربوط) دیتا را کاهش می دهد. مثلاً در مثال تشخیص عکس سگ از گربه، برای شناسایی حیوان خانگی مورد علاقه خود، نیازی به تصویر 4k با وضوح بالا که بتوانید روی یک موبایل کنیم نداریم. یک تصویر پیش نمایش کوچک، یا حتی یک تصویر کوچک، اغلب برای تشخیص اشکال متمایز کافی است و AE جزئیات غیرضروری را حذف میکند.

## ۲. مشکلات یادگیری بدون نظارت (Unsupervised Learning Problems)

می‌توانیم از AE برای یافتن کلاس‌های جدید استفاده کنیم، اگر اطلاعات ما هنوز برچسب‌گذاری نشده باشد. با کاهش داده‌های خود به یک زیرمجموعه بسیار کوچک (یعنی ۲ یا ۳ نورون خروجی)، مدل را آموزش می‌دهیم تا یاد بگیرد که چگونه داده‌ها را می‌توان به طور مؤثر به کلاس‌های فرعی تقسیم کرد. این برای یادگیری بدون نظارت یا خوشه بندی مفید است.

## ۳. تشخیص ناهنجاری (Anomaly Detection)

یکی از راه‌ها و تکنیک‌ها برای تشخیص ناهنجاری‌ها و موارد پرت AE ها هستند.

## ۴. Image Denoising :

حذف نویز یا کاهش نویز فرآیند حذف نویز از سیگنال است که می‌تواند یک تصویر، صدا یا یک سند باشد. می‌توانیم یک شبکه Autoencoder را آموزش دهیم تا نحوه حذف نویز از تصاویر را بیاموزید.

---

## سوال ۳:

خیر ، نمی‌تواند تصویر را به طور کامل بازسازی کند.  
زیرا شبکه ارائه شده overcomplete هست. یعنی ابعاد bottleneck بزرگتر از ابعاد لایه input می‌باشد. و در این حالت فرآیند یادگیری ممکن است در یادگیری مفید به مشکل بر بخورد.

---

## سوال ۴:

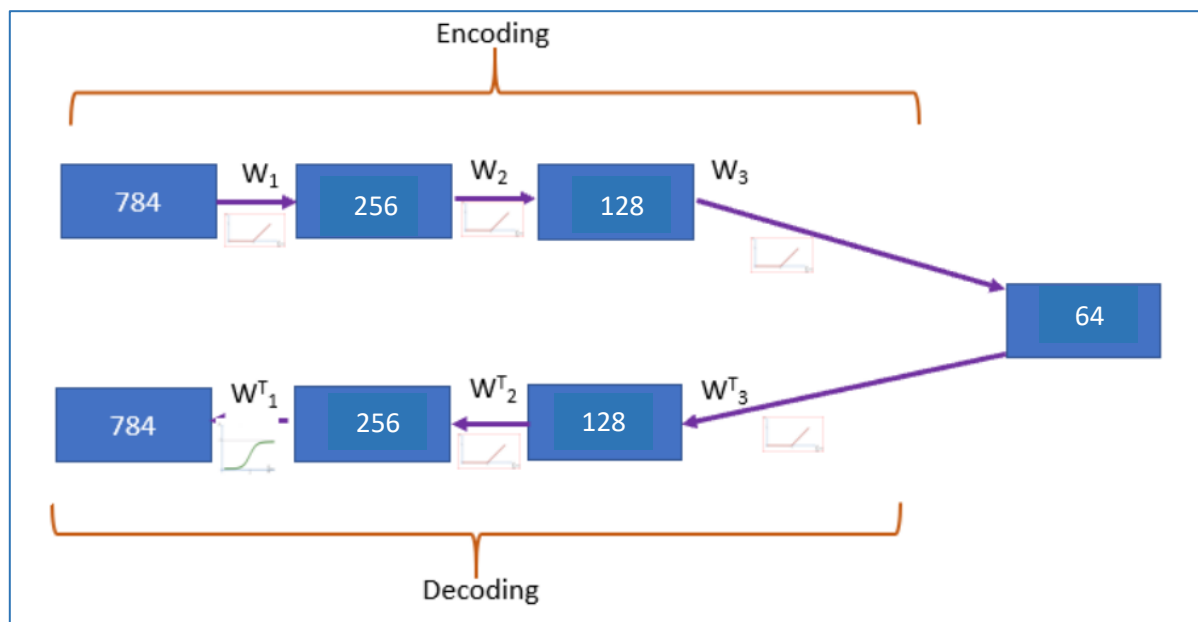
Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[None, 784]	0
dense (Dense)	(None, 256)	200960
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 128)	8320
dense_4 (Dense)	(None, 256)	33024
dense_5 (Dense)	(None, 784)	201488

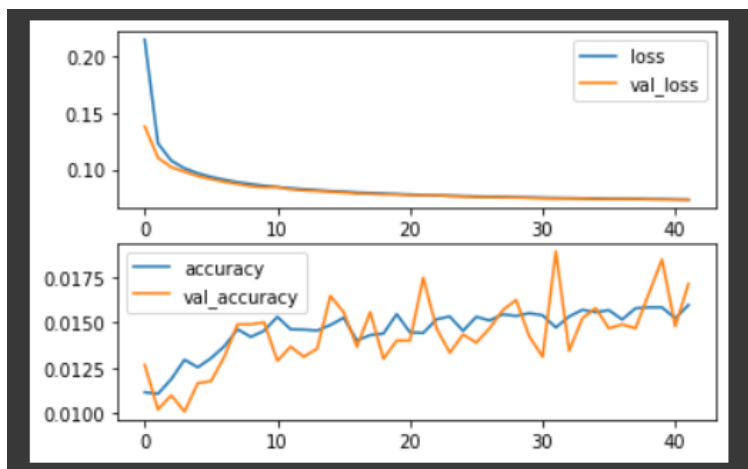
=====  
 Total params: 484,944  
 Trainable params: 484,944  
 Non-trainable params: 0

(۱) تعداد پارامترهای این مدل با ۵ لایه dense ، ۴۸۴۹۴۴ می باشد.

(۲)



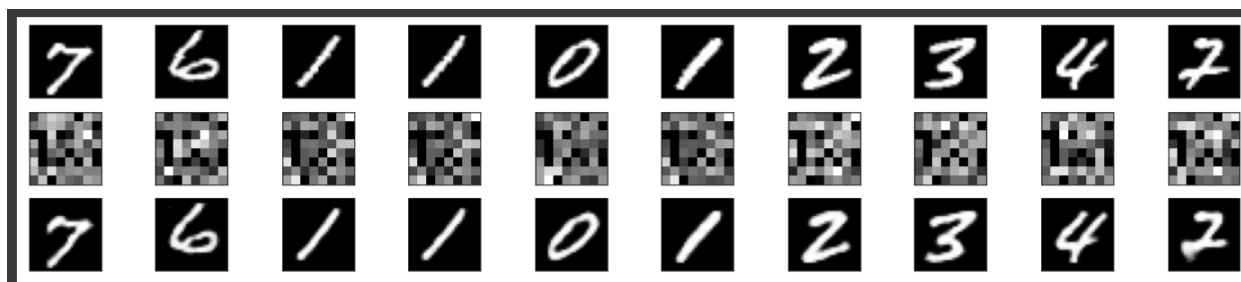
قسمت encoding ، لایه ورودی با ۷۸۴ واحد، و لایه های dense بعدی به ترتیب ۲۵۶ و ۱۲۸ واحد و با تابع فعالسازی relu تعریف شده است.  
 قسمت code یا bottleneck با ۶۴ واحد و تابع فعالسازی relu تعریف شده است.  
 قسمت decoding هم مانند encoding است فقط در لایه dense آخر تابع فعالسازی sigmoid می باشد.



(۳) برای validation set از داده های تست استفاده کردیم. ۹۰٪ را برای validation و مابقی را برای آزمون قرار دادیم. نمودار خطا و دقت داده های آموزش و validation در نمودار روبرو

قابل مشاهده است. خطای مدل سیر نزولی و رو به صفر داشته و دقت مدل به صورت نوسانی سیر صعودی داشته. نوسانی بودن به دلیل وجود epoch و batch\_size است که shuffle انتخاب میشوند.

(۴)



سطر اول تصویر موجود در دیتاست هست. سطر دوم لایه code یا bottleneck می باشد و سطر سوم تصویر بازسازی شده توسط مدل اتوانکدر تعریف شده.

(۵)

لایه میانی مدل ما ۶۴ نورون دارد. همین مدل را با تعداد نورون کمتر در لایه میانی اجرا کردیم و براساس نتایجی که بدست آمد، تعداد نورون بیشتر در لایه میانی بازدهی مدل را افزایش می دهد.

## سوال ۵:

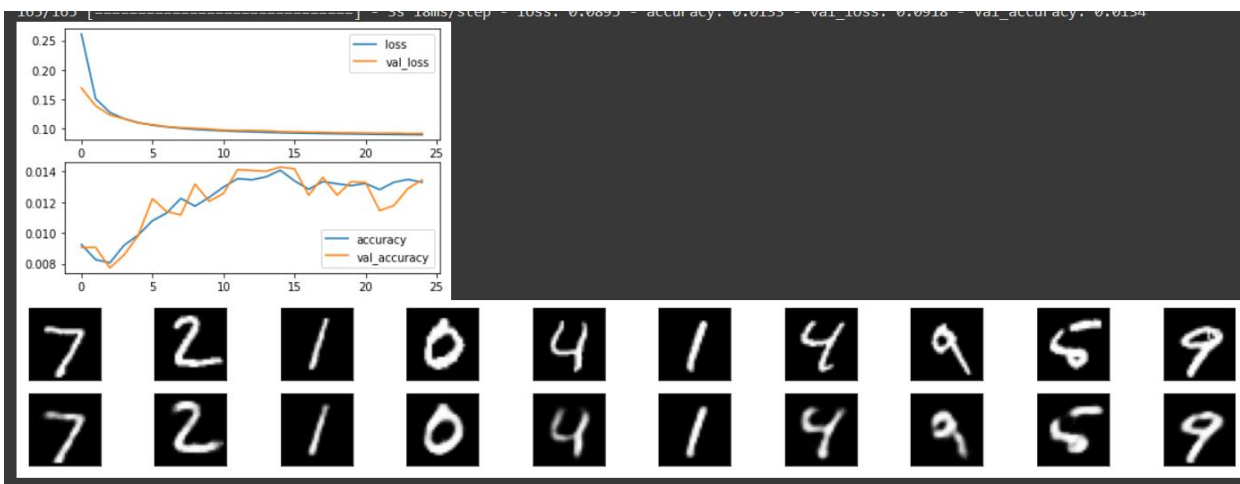
(۱)

```
#adding noise
factor = 0.42
X_train_noisy = x_train + factor * np.random.normal(loc = 0.0, scale = 1.0, size = x_train.shape)
X_train_noisy = np.clip(X_train_noisy, 0., 1.)
X_test_noisy = x_test + factor * np.random.normal(loc = 0.0, scale = 1.0, size = x_test.shape)
X_test_noisy = np.clip(X_test_noisy, 0., 1.)
```

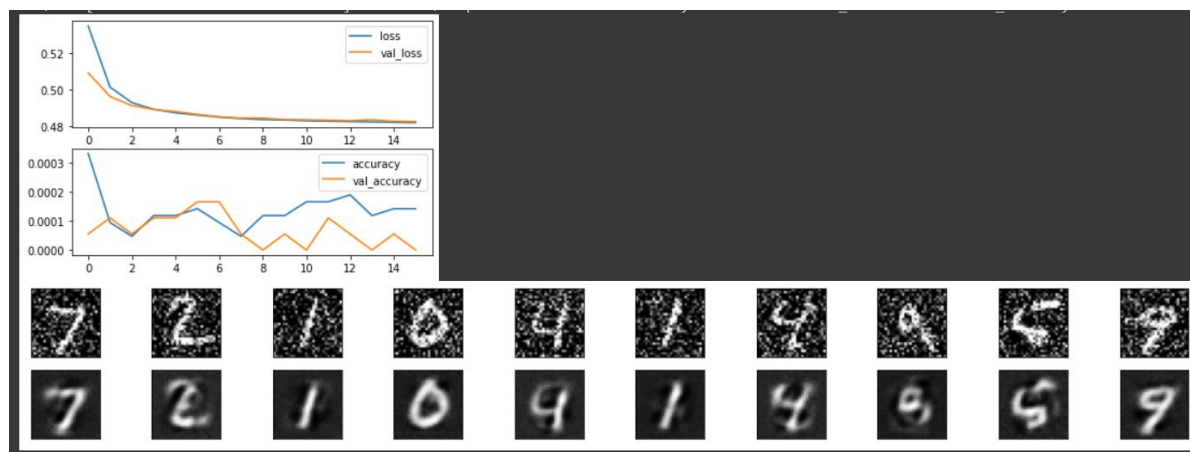
با استفاده از تابع `random` به تصاویر اولیه رندوم اضافه میکنیم که باعث ایجاد نویز در تصویر میشود. عامل `factor` را هم برای کنترل میزان نویز تعریف میکنیم که هرچه بزرگتر باشد نویز تصویر بیشتر میشود. در نهایت چون حداکثر مقدار پیکسل برای تصاویر ما ممکن است از ۱ تجاوز کند، بنابراین باید تصاویر را با دستور `clip` برش می دهیم.

(۲) مدلی با ۳ لایه `dense` تعریف کردیم که قسمت `encode` آن ۱۲۸ نورون با تابع فعالسازی `relu`، قسمت کد ۳۲ نورون با تابع فعالسازی `relu` و قسمت `decode` آن ۱۲۸ نورون با تابع فعالسازی `sigmoid` است.

این مدل را روی داده های اصلی که نویز به آنها وارد نشده ، اعمال کردیم و نمودار دقت و خطای این مدل به صورت مقابل است. خطا کاهشی و دقت سیر صعودی دارد. در قسمت دوم تصویر، سطر اول داده های اصلی و سطر دوم داده های بازسازی شده توسط مدل را نشان می دهد.



(۳) همان مدل قسمت ۵-۲ را این بار روی داده های نویزی کامپایل کردیم. همانطور که در نمودار دقت و خطا مشاهده میشود. این مدل برای داده های نویزی دارای خطای بسیار زیاد و دقت خیلی کم نزدیک به صفر است. یعنی این مدل برای داده های نویزی عملکرد خوبی ندارد.



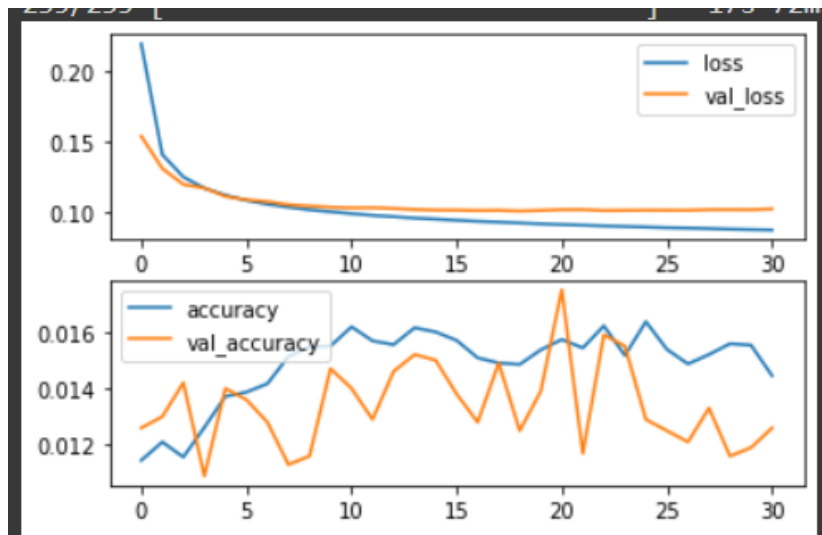
(۴)

```
# Q 5-4
#define autoencoder for denosing

#input layer
input_img = Input(shape=(784,))
#encoding architecture
encode_layer1=Dense(units=800 , activation='relu')(input_img)
encode_layer2=Dense(units=400, activation='relu')(encode_layer1)
#latent view
latent_view=Dense(units=50 , activation='relu')(encode_layer2)
#decoding architecture
decode_layer1=Dense(units=400 , activation='relu')(latent_view)
decode_layer2=Dense(units=800 , activation='relu')(decode_layer1)
#output layer
output_img=Dense(units=784 , activation='sigmoid')(decode_layer2)
```

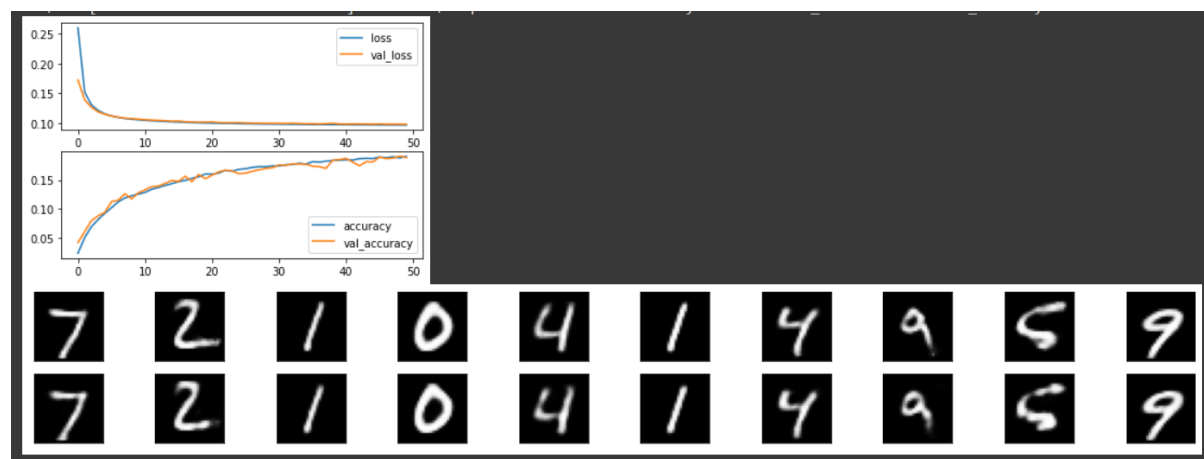
معماری اتوانکدر تعریف شده برای حذف نویز به این صورت می باشد. لایه اول و دوم انکدر به ترتیب ۸۰۰ و ۴۰۰ نورون با تابع فعالسازی relu ، لایه میانی با ۵۰ نورون و تابع فعالسازی relu ، لایه

اول و دوم دیکدر به ترتیب ۸۰۰ و ۴۰۰ نورون با لایه فعالسازی relu و خروجی ۷۸۴ نورون با تابع غیرخطی sigmoid. نتایج حاصل از این مدل که برای حذف نویز استفاده میشود در زیر قابل مشاهده است.



(۵)

مدل قسمت ۲-۵ را روی داده هایی که خودمان حذف نویز کردیم ، اعمال کردیم. نتایج آن به صورت زیر است. همانطور که مشاهده میشود، خطا سیر نزولی دارد و دقت با شیب خوبی در حال افزایش است. در قسمت دوم تصویر ، سطر اول داده های حذف نویز شده هستند و سطر دوم داده هایی که توسط مدل ساده با ۳ لایه dense بازسازی شده اند.



(۶) سطر اول داده های اصلی آموزش، سطر دوم داده های نویزی شده ، سطر سوم داده های حذف نویز شده توسط اتوانکدر تعریف شده، می باشد.

5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9

(۷) سطر اول داده های اصلی آزمون، سطر دوم داده های نویزی شده ، سطر سوم داده های حذف نویز شده توسط اتوانکدر تعریف شده، می باشد.

7	2	1	0	4	1	4	9	5	9
7	2	1	0	4	1	4	9	5	9
7	2	1	0	4	1	4	9	5	9

(۸)

ما یک مدل ثابت را روی سه نوع دیتای مختلف اعمال کردیم. برای داده های اصلی موجود در دیتاست، برای داده هایی خودمان به انها نویز وارد کردیم، و برای داده های نویزی شده که با اتوانکدر نویز آنها را حذف کردیم.

با توجه به میزان دقت و خطا و تصاویر بازسازی شده برای هر سه نوع دیتا ورودی که به تفکیک در قسمت های قبلی گزارش آمده است، میتوان نتیجه گرفت که هر چی نویز تو داده ورودی مون کمتر باشه مدل میتونه تصاویر بهتر و نزدیکتر به تصویر اصلی را بازسازی کند. چون میزان دقت مدل در داده هایی که حذف نویز شده بودند به نسبت ۲ حالت قبل افزایش زیادی داشت. همچنین میتوان نتیجه گرفت که اتوانکدري که برای حذف نویز استفاده شده کارامد بوده و توانسته به خوبی نویز ها را حذف کند و تصویر پردازش شده ی خوبی را بعنوان ورودی برای مدل دهد.



## سوال ۶:

۱- رمزگذار خودکار متغیر را می توان به عنوان یک رمزگذار خودکار تعریف کرد که آموزش آن برای جلوگیری از **overfitting** منظم شده است و اطمینان حاصل می شود که فضای پنهان دارای ویژگی های خوبی است که فرآیند تولید را امکان پذیر می کند. علاوه بر این، اصطلاح "variational" از رابطه نزدیکی که بین روش منظم سازی و استنتاج متغیر در آمار وجود دارد، می آید.

شبکه عصبی خود رمزگذار متغیر (VAE) در مقایسه با شبکه عصبی خود رمزگذار (AE)، احتمالات را به جای ویژگی ها فشرده می کند. علاوه بر تغییرات کوچک که بین دو شبکه عصبی مذکور اتفاق افتاده است، هر یک از این انواع شبکه های عصبی مصنوعی به پرسش متفاوتی پاسخ می دهند. شبکه عصبی خود رمزگذار به پرسش «چگونه می توان داده ها را تعمیم داد؟» پاسخ می دهد، در حالی که شبکه عصبی خود رمزگذار متغیر به پرسش «اتصال بین دو رویداد چقدر قوی است؟ آیا باید خطا را بین دو رویداد توزیع کرد یا آن ها به طور کامل مستقل هستند؟» پاسخ می دهد.

-۲

