**CS5542 Big Data Apps and Analytics**

**In Class Programming –8 Report**

**(Jongkook Son)**

# Project Overview:

**Use the same data and source code but add two more layers to encoder path and their corresponding two layers to decoder path, run the new model and report your findings. In your report specify which 4 layers (2 layers in encoder path and 2 layers in decoder path) have you added and explain why you added those (their function).**

**Examples of layers that can be added Conv2D, Batchnorm, Conv2DTranspose etc.**

# Requirements/Task(s):

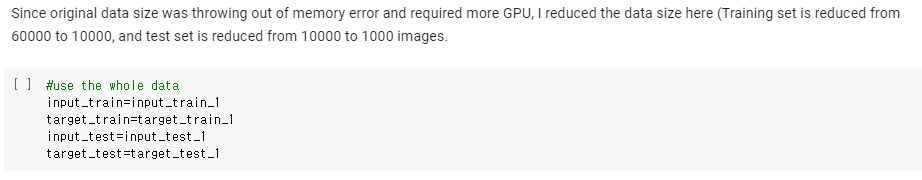
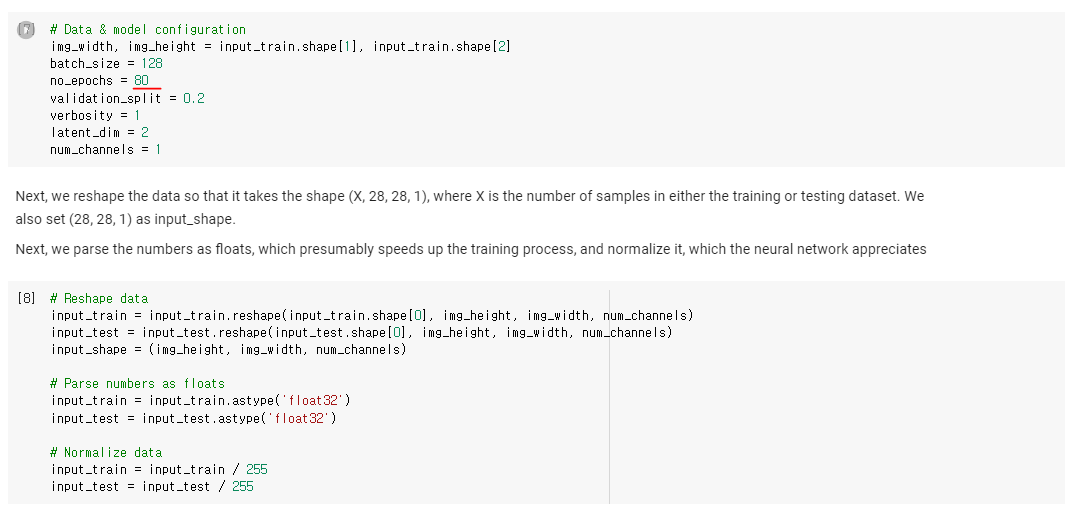
1. Successfully executing the code with new architecture for encoder and decoder path (75 points)
2. Explanation of new layers (5 points)
3. overall code quality (10 points)
4. Pdf Report quality, video explanation (10 points)

**What I learned in ICP:**

I could have learned the basic structure of the encoder and decoder and variational auto encoders by doing this ICP. Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. So in our icp I tried to make encoder and decoder with asymmetric structure. Like adding Conv2dTranspose to decoder if I add a Conv2d to encoder. The whole training process is like Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data. In other words, they are self-supervised because they generate their own labels from the training data. In this ICP I used a picture datas and by using autoencoders could genereate own labels of data and could have generated new pictures from it. Finally it is unavoidable that there could be some loss in auto encoders so the generated picture is not 100 percent same with the original picture.

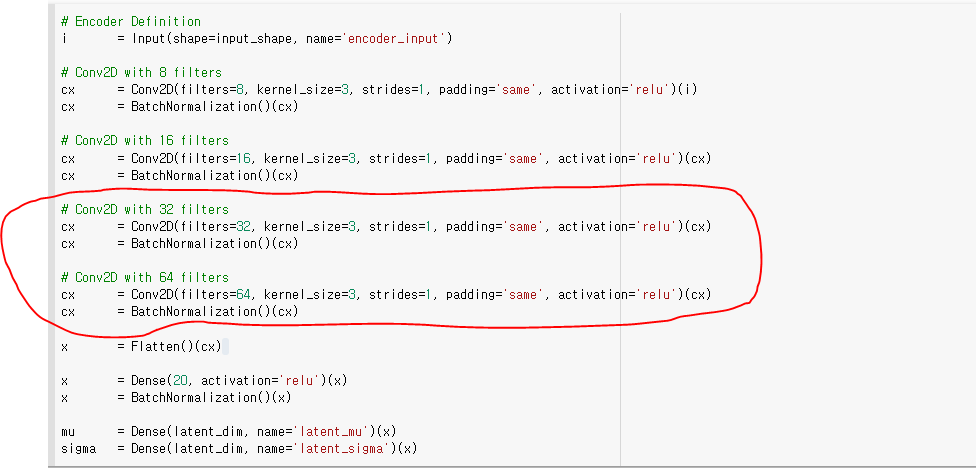
**ICP description what was the task you were performing and Screen shots that shows the successful execution of each required step of your code**

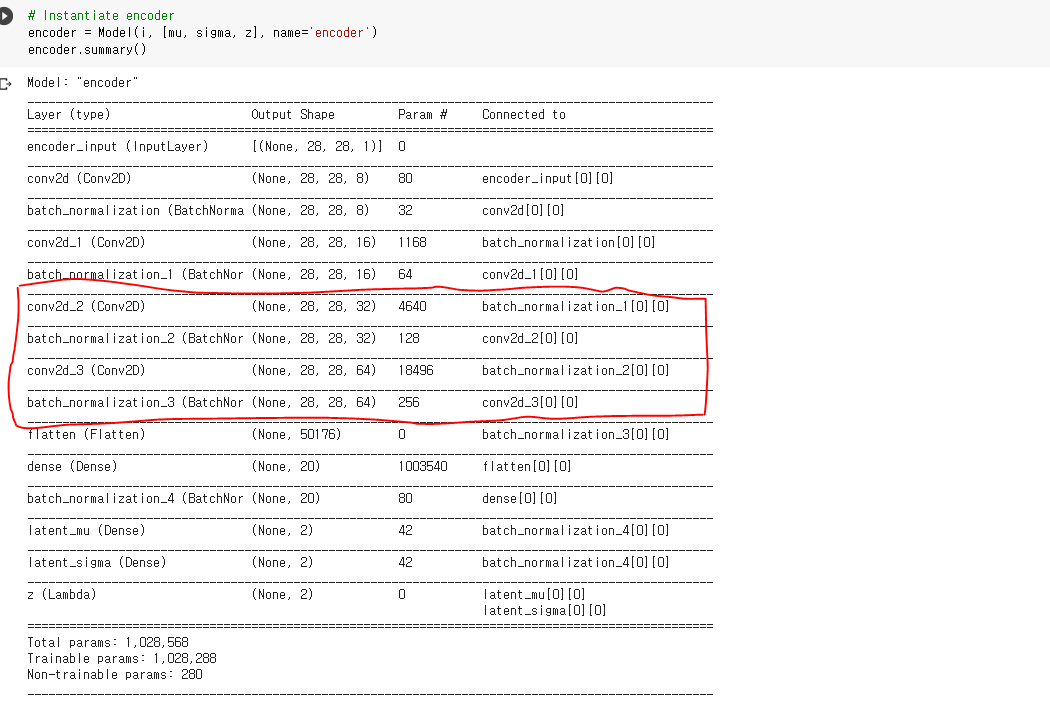
**Configuration**

****

* **In the source code, training set and test is reduced due to memory issue. However in my ICP I tried to use the whole data since my local environment can handle the datas. Also I increased the number of epoch 80 to get more accurate result.**

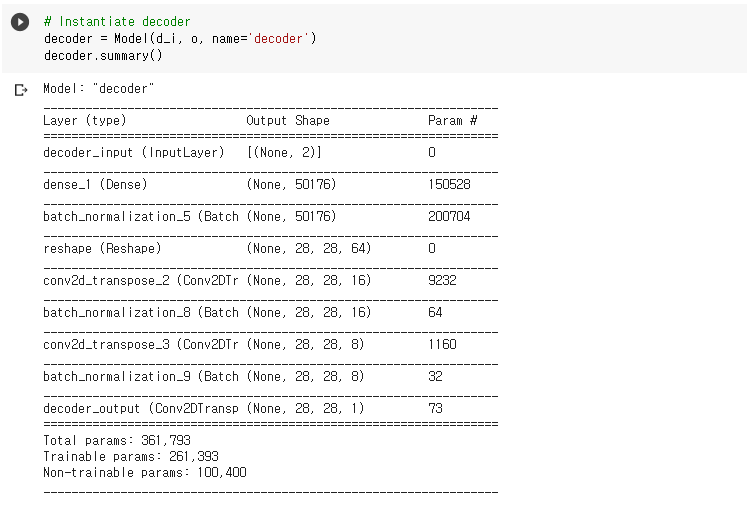
**ADD 4 layers in encoder path and 4layeres in decoder path**





* **I add 4 layers in my ICP. Which are Conv2d layers with 32 filters 64 filters. The reason I add these layers is that by adding increased filter I can train more complex and deeper pattern of the picture. Also I added batch normalization layer after each Conv2D layers. This layer ensures that the outputs of the Conv2D layer that are input to the next Conv2D layer have a steady mean and rate and makes overall training faster and steady.**

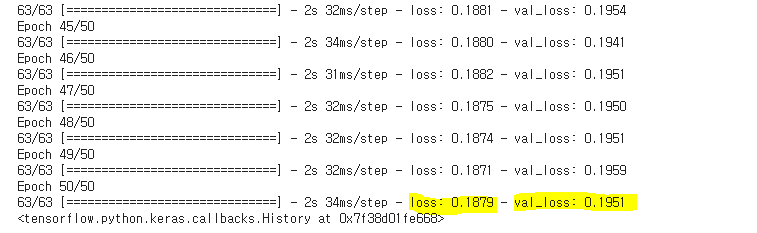




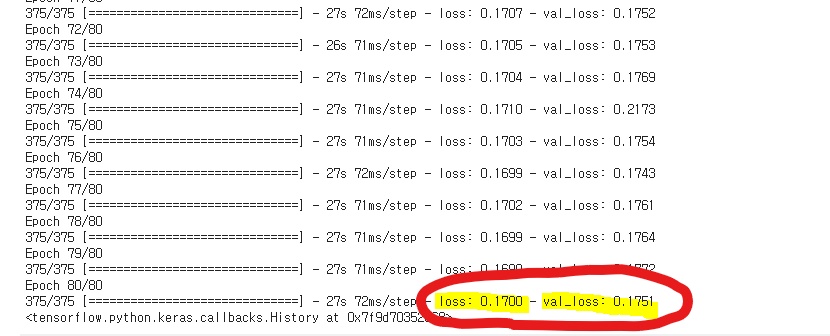
* **Since we added Conv2d layers with 32 filters and 64 filters. We need to add Conv2DTranspose and BatchNormalization in the exact opposite order as with our encoder.**

**Result of the ICP**

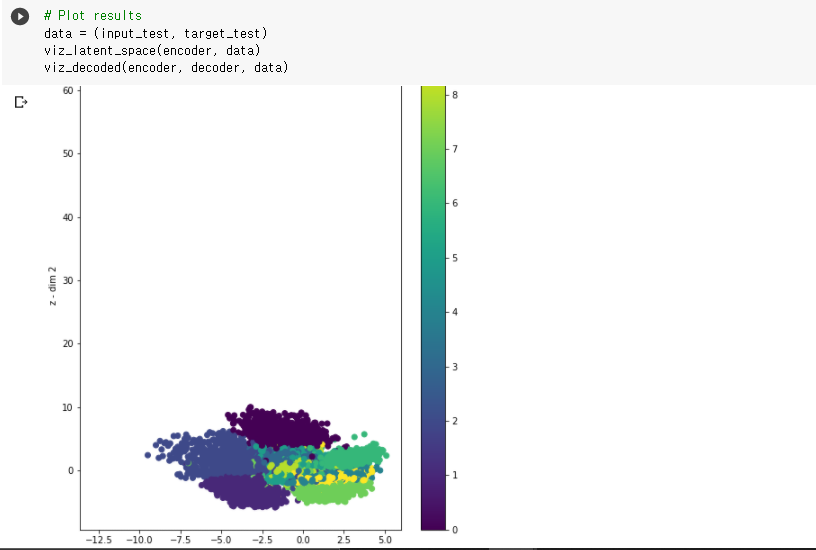
**<The training result of the Source Code>**

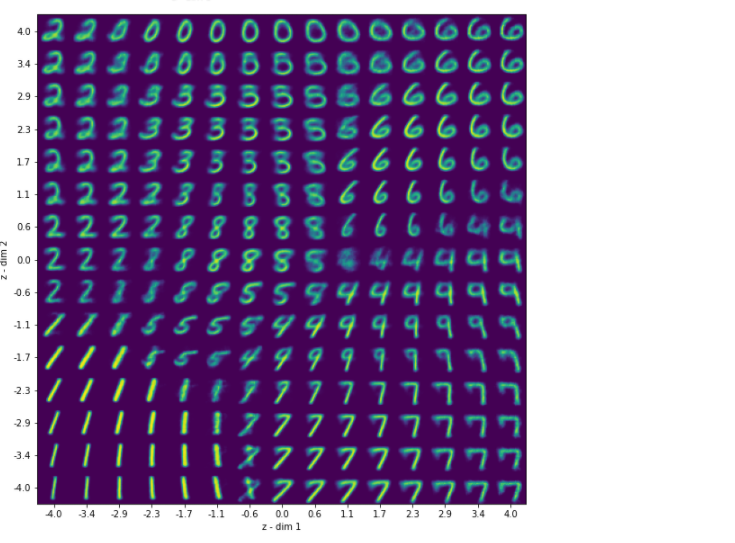
****

**<The training result of the ICP>**

****

**<Visualization Result>**

****

****

* **The loss and validate loss number for the training is getting better in this ICP. Even though there are some loss in the training process the overall visualization result is quite good.**

# Conclusion:

# By adding a Conv2d layers with 32 filters and 64 filters and batch normalization. More complex and deeper patterns of the pictures in the dataset is trained. So the loss and val\_loss number is decreased and overall training result is getting better. And the visualization result is getting better compared to Source Code.

# Challenges that I faced:

The most difficult challenge that I faced was that it was hard to grasp the structure of the variational autoencoders. And especially hard to understand how encoder and decoder is connected in this VAE. But I figured out there is an process named reparameritization in this process we can reparameterize the sample fed to the function into the shape μ+σ2×ϵ, it now becomes possible to use gradient descent for estimating the gradients accurately the sampled z values and feed it to the decoder, which ensures that we arrive at correct VAE output. And this makes encoder and decoder connected and makes gradient decent work more efficiently.

# Video link

<https://www.youtube.com/watch?v=rXc9OlKBnw8>