You are provided with three notebooks: 17\_autoencoders\_and\_variational\_autoencoders.ipynb written by Aurelien Geron for the 2nd Edition of his book and chapter12\_part04\_variational-autoencoders.ipynb for the 2nd edition of Chollet’s book and convolutional\_variational\_autoencoder.ipynb from the TensorFlow site. Please save unmodified copies of each notebook so you could compare the original results with your own.

**Problem 1.**  Build a denoising convolutional autoencoder using Keras API. Create Train, Validate and Test sets of images. Demonstrate that noise is successfully removed by passing several images of handwritten digits: 1,2,5,9, and 4, with added noise. Choose those images from the Test set and not the Training dataset. You are expected to use images from the MNIST dataset. Please plot the original images, noisy images and reconstructed images.

Experiment with several sizes of the middle (coding) layer. For example, use sizes: 64,32, 16, and 8 Compare qualities of output (reconstructed) images as a function of coding layer size.

In each case, capture and present coding vectors for numbers 1,2,5,9, and 4 as rectangular images. Make qualitative and quantitative comparisons of the qualities of reconstructed images. Explain how you are doing it. Provide verbal description of those qualities and quantitative measures. Perhaps you want to place output (reconstructed) images of separate digits for different codings sizes one next to another. That would allow an easy visual comparison. Present the working code. **(25%)**

**Problem 2.** Use whichever codings size in the previous problem you discover is the best and create a new autoencoder with Conv2D and MaxPol2D layers rather than Dense layers. Compare the accuracy of the new model and the quality of reconstructed images for the digits 1,2,5,9, and 4 used in problem 1. Report your findings. **(20%)**

**Problem 3.** Starting with the notebook: convolutional\_variational\_autoencoder.ipynb construct a Variational Autoencoders (VAE) using Conv2D and Conv2DTtranspose layers. Experiment with the latent space of dimension 2 and 8. Use the same Train, Validate and Test sets of images used in Problem 1. Train your VAEs for those two latent space dimensions. Once the training is done, take the same images of digits 1,2,5,9, and 4 used in the Problem 1. Find the outputs of trained VAEs with two different dimensions of the latent space for those images. Compare those outputs with the outputs obtained in the first problem by the autoencoder. We are just curious. Is one latent space “better” than the other? **(25%)**

**Problem 4.** Use one of the VAEs from the previous problem. Choose the one you consider “better”. Tell us which criteria you used for the quality of the VAEs. Identify mean codings for four digits 1,2,5,9, and 4 from the test set. You could use the same images as in previous problems. Use the encoder portion of your VAE to generate the mean coding for all four digits (images). Imagine a “line” between coding vectors for images 2 and 5 and similarly images 1 and 4. On each of those two lines place 8 equally distant points. In the variational autoencoder the middle output (codings) has two components: the  mean coding and the standard deviation . Use the mean codings of the same image of numbers 2 and number 5. Each is a 2 or 18 dimensional vector, call them a and b. You make a sequence of vectors with 8 intermediate values by generating vectors: a, a + (b-a)/9, a + 2\*(b-a)/9,  a + 3\*(b-a)/9,...., 8\*(b-a)/9, b. For each of those, use the decoder to predict the reconstructed image. Use the decoder portion of your VAE to predict the images corresponding to those points. In that “prediction” include the boundary images 2 and 5 and 1 and 4 respectfully. **(30%)**

Please add your name to the top of your notebooks.

We expect you to submit one notebook (\*.ipynb file) along with its HTML images.

These problems might require GPU processing. Unless you have a decent GPU card yourself, do those problems in Google Colab.

If your notebook(s) contain(s) excessively long outputs please copy a meaningful and illustrative number of initial and/or final lines and paste those in a markdown (comment) cell. Then, delete the long output(s).