

# Generative Models: Project 1

Hongli Lin (kag52rer), Michele Paterson (mipat104),  
Jose Leal Miudo (slea100), Sven Klein (svkle100)

November 25, 2024

## Exercise 1:

Solutions to this exercise are in `exercise1.ipynb` in the notebooks folder. We decided for B20 as the test brain since all brains seem to have an equal distribution of image sizes and B20 has the least amount of images. Therefore we lose the least amount of training data by picking B20 for testing. The minimum, maximum, mean and standard deviation are stored in a json file called `stats.json`. We also created some util methods in the `data_util` module which help with loading and visulizing the dataset.

## Exercise 2:

Solutions to this exercise are in `exercise2.ipynb` in the notebooks folder. The classes used for sampling and storing the data are in the `data_utils` module. Brain Dataset is a Dataset that has all the methods needed to store the brains. Then the class Brain Sampler is used the get random brain images. For this it has two methods. If you set weighting to equal, then it will pick a random brain with equal probability and then pick a random image of that brain again with equal probability. If you set the weighting to `by_pixel` it will select the images such that each pixel has an equal probability to be seen by our VAE. This should in Theory make better use of our training data.

## Exercise 3:

Solutions to this exercise are in `exercise34.ipynb` in the notebooks folder. The implementation of the model can be found in the `vae` module. We choose a normal Distribution with mean 0 and variance 1 for our latent space distribution. We can calculate the KL divergence between two normal distributions by  $KL(n_1, n_2) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$ . Since we have  $\sigma_2 = 1$  and  $\mu_2 = 0$  this simplifies to  $KL(n_1, n_2) = -\frac{1}{2}(1 + \log(\sigma_1^2) - \sigma_1^2 - \mu_1^2)$ . To avoid issues where  $\log(\sigma_1)$  might not be defined we decide to learn  $\log(\sigma_1^2)$  instead of  $\sigma_1$ . For the model we use convolutional layers with 1 fully connected layer to obtain  $\sigma_1$  and  $\mu_1$ . The model is also inspired by the architecture used in Higgins et al. [2017], but was made smaller for faster

training.

#### **Exercise 4:**

Solutions to this exercise are in `exercise34.ipynb` in the notebooks folder. The trained model and its parameters can be found in the models folder.

#### **Exercise 5:**

Solutions to this exercise are in `exercise5.ipynb` in the notebooks folder. The images generated by the model look like plausible brain structures at least to non experts. The Interpolation between tiles looks very clean and you can see the cell structures slowly deforming to transition from one image to another. Looking at the PCA in 2 dimensions the latent space looks kind of like an ellipse. Overall the latent space has to be fairly smooth and interpretable, otherwise the interpolation wouldnt be possible to this extent.

## **References**

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Sy2fzU9gl>.