DLE Homework 7
Gaussian Variational Autoencoders

Vojtěch Michal, michavo3

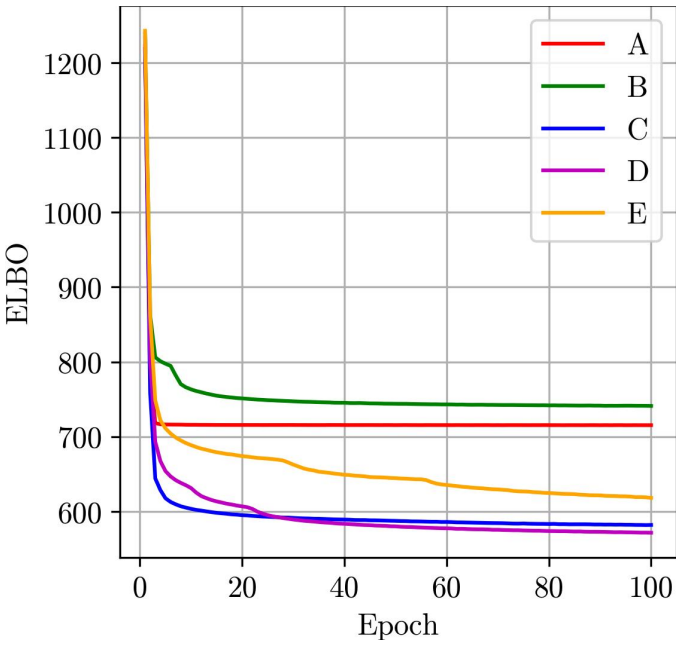

Fig. 1: nELBO curves recorded during the training of individual VAEs

***Abstract*—The assigned homework[1] focuses on the capability of Gaussian Variational Autoencoders (VAEs) to generate new objects from $\mathcal{X}$ (in our case the MNIST dataset of 28x28 pixel handwritten digits) using much less dimensional latent variables from $\mathcal{Z}$ (in our case). Several VAEs of different architectures and complexities were trained to minimize the negative evidence lower bound (nELBO) on 60000 examples from the MNIST, all with the dimension 12 of latent space $\mathcal{Z}$. The report presents reconstruction (encoding and subsequent decoding of) images by each VAE as well as the generation of new images from randomly generated latent variables. Finally, the limiting distribution of individual VAEs is analyzed.**

I. Implementation of VAE

The decoder and the encoder are feedforward neural networks composed of fully connected linear layers and ReLU activations. The dimensionality of images from $\mathcal{X}$ is fixed at 784 due to the image resolution. The author of [1] suggests that the exact $\dim\mathcal{Z}$ does not matter much in a wide range of values. Specifically for the case of MNIST, he mentions that $\dim\mathcal{Z} <= 4$ yields bad results, hence latent code was chosen to be of dimension 12.

[1]The homework assignment is available on https://cw.fel.cvut.cz/wiki/courses/bev033dle/labs/lab7_vae/start

Model	Number of parameters	Encoder hidden layer widths
A	29033	[]
B	52201	[32]
C	103593	[64]
D	100065	[60, 32]
E	100953	[60, 32, 20]

TABLE I: Overview of individual tested models, and the number and organization of their parameters.

Model	Final nELBO
A	719.1
B	746.8
C	579.1
D	571.6
E	613.5

TABLE II: Final values of the negative ELBO criterion achieved by individual models during training.

Five different models were implemented and tested to assess the influence of the model's complexity on its capabilities. Each has an encoder composed of an input layer of 784 units, an output linear layer of 24 units (corresponding to twice the dimension of $\mathcal{Z}$) and a varying number of hidden layers of various number of units listed in Table I. Each hidden linear layer is followed by a ReLU activation function. Transformation from $\mathcal{Z}$ to $\mathcal{X}$ is implemented by the decoder with the same structure reversed.

II. Training

Each model was trained for 100 epochs using the Adam optimizer with learning rate $l_r = 0.001$. Learning was performed using mini batches of 64 samples. The recorded decrease of negative ELBO criterion averaged over the training set is shown in Fig. 1 with the final values for each model listed in Tab. II. Fig. 2 shows the evolution of the KL divergence term $D_{KL}(q(z|x)||p(z))$ penalizing the "distance" of encoder distribution $q(z|x)$ from prior distribution $p(z)$ of z that is the standard normal distribution $\mathcal{N}(0, \mathbb{I})$.

To test how well each model can close the loop, i.e. start with an image $x \in \mathcal{X}$, encode it into a probability distribution $q(z|x)$ in $\mathcal{Z}$, sample z from it and reconstruct an image $\hat{x} \in \mathcal{X}$ similar to the initial x, a batch of 16 random images from the test set of MNIST (i.e. previously unseen images) was passed through each VAE with results presented in Fig. 4. The leftmost column contains the original images. The second column shows the baseline VAE A with no hidden layers that yields the most blurry result. The sharpest reconstruction is achieved by models C and D (fourth and fifth columns), which also reached the lowest value of the nELBO criterion.

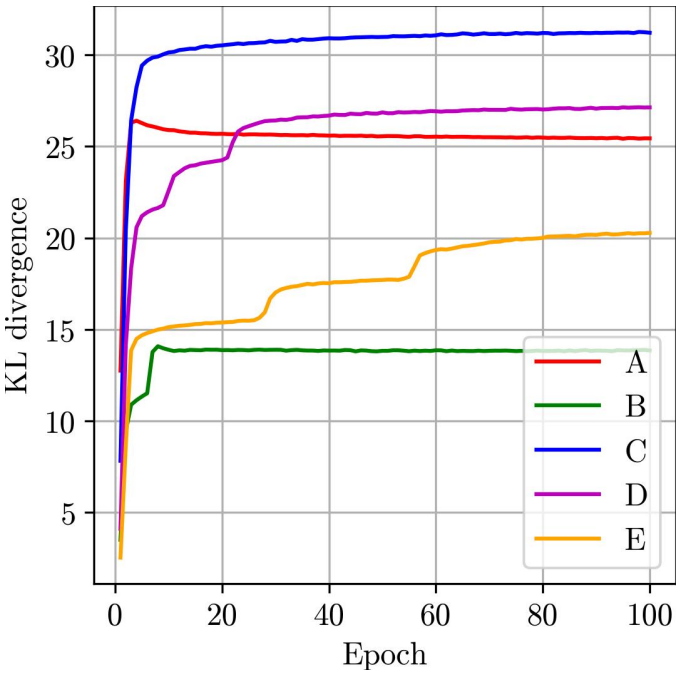

Fig. 2: KL Divergence recorded during the training of individual VAEs

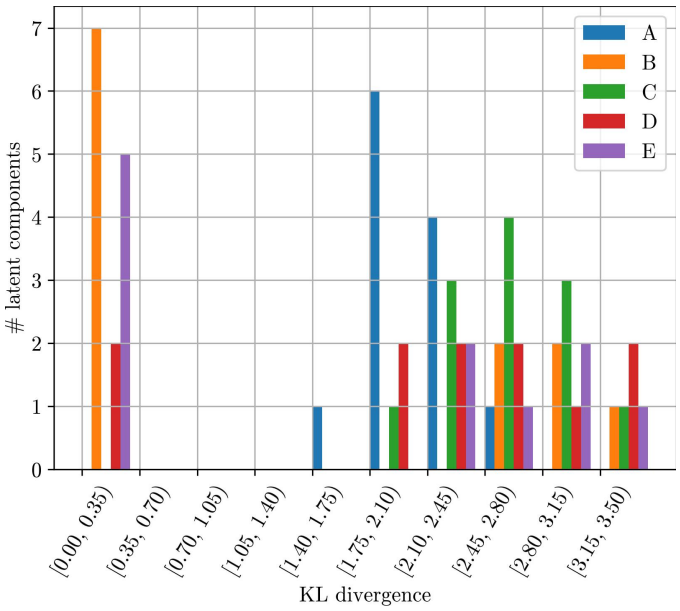

Fig. 3: Histogram of average values of latent variables on a batch of 256 images.

III. Evaluation of VAEs

A. Posterior collapse

It is interesting to compare the number of latent code components $z_i \in \mathbb{R}$ that are actually used by each VAE. A batch of 256 MNIST test set images waw encoded into individual distributions $q_j(z|x_j)$ on $\mathcal{Z}$. The expression $D_{KL,j}(q_j(z|x_j)||\mathcal{N}(0, \mathbb{I}))$ was averaged over all images in the batch and plotted as a histogram in Fig. 3. Components that fall into the bin close to zero have probability distribution

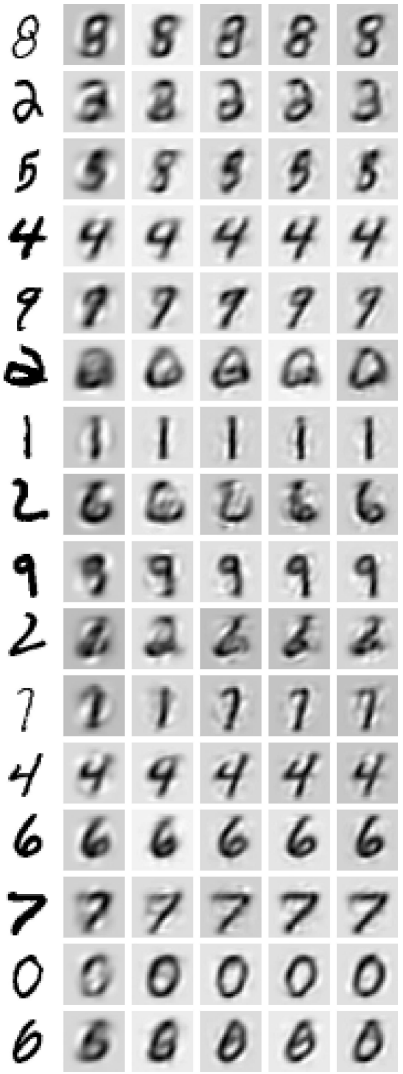

Fig. 4: Examples of images reconstructed using resampled latent codes. From left to right: original images, VAE A through E.

$q(z|x)$ very close to $\mathcal{N}(0, \mathbb{I})$ and therefore carry essentially no value for the model. There is a lot of such collapsed components in the case of models B and E that achieved the worst performance w.r.t. the nELBO criterion shown in Fig. 1.

B. Generating random images

To evaluate the behaviour of decoder mapping from latent codes $\mathcal{Z}$ to images $\mathcal{X}$, a batch of 60 random latent codes was drawn from $\mathcal{N}(0, \mathbb{I})$ and fed through each VAE decoder. Obtained images are shown in Fig. 5 through 9.

Images generated this way by the baseline model A are the most blurry with least resemblance of any handwritten digit. Similarly, images generated by VAE C and D are also very blurry and more often than not resemble no digit as well. This is, however, expected, as Fig. 2 shows that the probability distribution of latent codes corresponding to meaningful images is far from standard normal $\mathcal{N}(0, \mathbb{I})$ that was used to draw z in this experiment. On the other hand, the model B has achieved significantly lower KL divergence term in the nELBO criterion and, therefore, random $z \sim \mathcal{N}(0, \mathbb{I})$ are closer to true latent codes corresponding to MNIST digit images. That is the reason why images generated by the model B look the best.

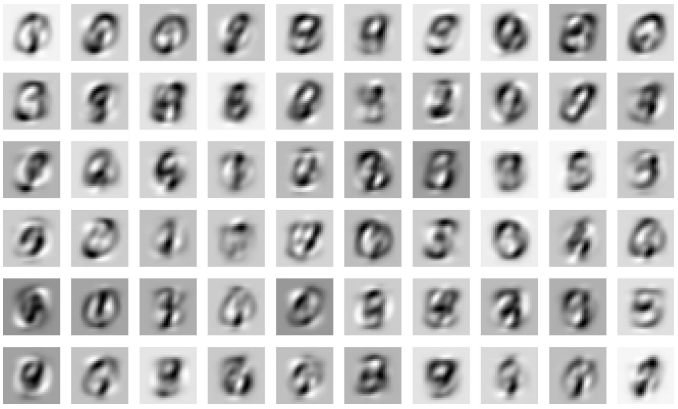

Fig. 5: Images decoded by model A from random latent codes $z \sim \mathcal{N}(0, \mathbb{I})$.

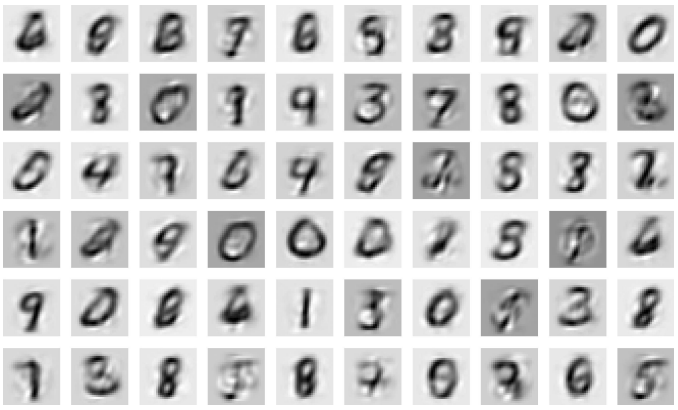

Fig. 6: Images decoded by model B from random latent codes $z \sim \mathcal{N}(0, \mathbb{I})$.

Fig. 7: Images decoded by model C from random latent codes $z \sim \mathcal{N}(0, \mathbb{I})$.

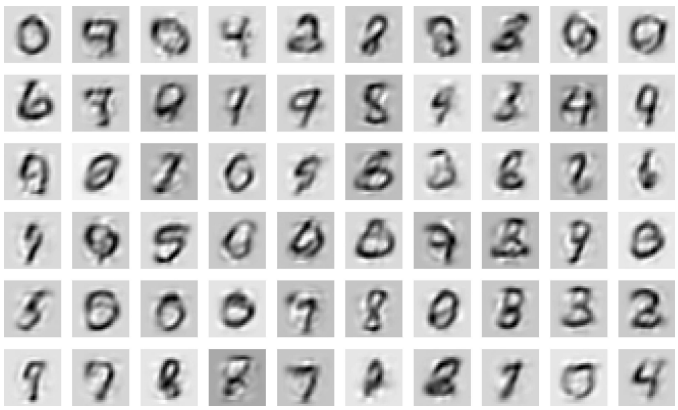

Fig. 8: Images decoded by model D from random latent codes $z \sim \mathcal{N}(0, \mathbb{I})$.

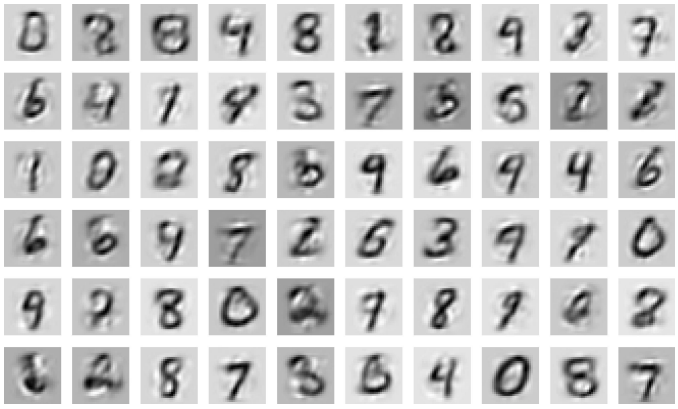

Fig. 9: Images decoded by model E from random latent codes $z \sim \mathcal{N}(0, \mathbb{I})$.

C. Limiting distributions

A fixed-point iteration was performed with each model to observe whether repeated encode-sample-decode-sample process will converge to some limiting probability distribution. The process started with a batch of 60 random latent codes sampled from $\mathcal{N}(0, \mathbb{I})$ and was run over the horizon of 100 iterations with results presented as animations attached in the submitted zip file. Final frames of each amination are shown in Fig. 10 through 14.

The model A (Fig. 10) is the best example of convergence towards a limiting distribution as all initially randomly selected latent codes converged to a single round shape resembling a mixture of digits 0 and 6. Similarly, the model C (Fig. 12) appears to have a limiting distribution on $\mathcal{X}$ that resembles a pointy shape of digits 1, 7 and 4.

The other models did not converge to any specific pattern, each image in the batch rather settled in some local fixed point and stopped charging after some number of iterations, but there is no ”global” fixed point for most of the batch.

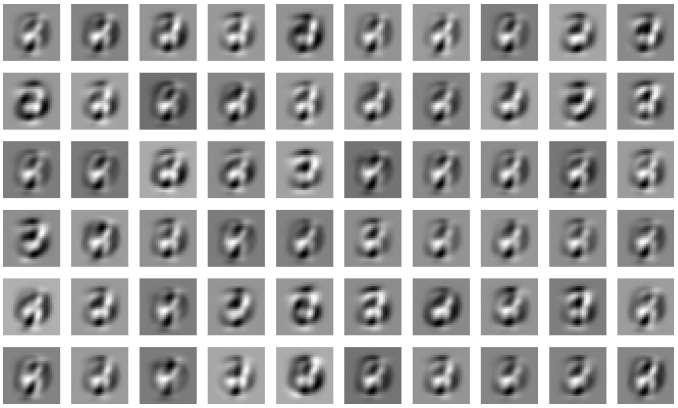

Fig. 10: Limiting distribution of model A after 100 iterations.

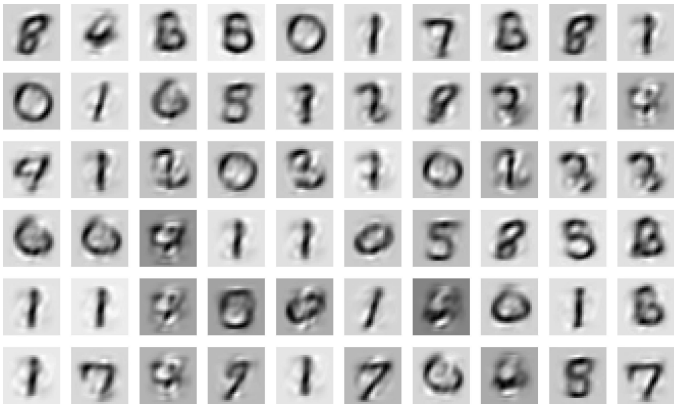

Fig. 11: Limiting distribution of model B after 100 iterations.

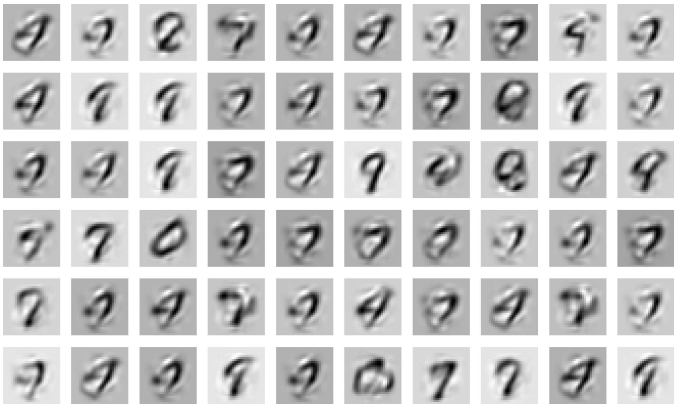

Fig. 12: Limiting distribution of model C after 100 iterations.

Fig. 13: Limiting distribution of model D after 100 iterations.

Fig. 14: Limiting distribution of model E after 100 iterations.

REFERENCES

[1] Doersch Carl, *”Tutorial on Variational Autoencoders”*, 2021, arXiv:1606.05908