Using Discrete VAEs on T1-Weighted MRI Data to Embed Local Brain Regions

Rohan Kapre
Mentored By: Jiahong Ouyang, Dr. Qingyu Zhao
CNS Lab
Stanford University
rokapre@stanford.edu

Abstract:

An issue with standard autoencoders (AEs) and variational autoencoders (VAEs) is that the latent space fails to capture local information in the input image. Capturing local information is important for distinguishing brain regions in T1 weighted MRI imaging. One method proposed in the past to address this is the vector quantized VAE (VQVAE), which uses a codebook to transform the encoder output into a discrete latent space through clustering. We find that while the VQVAE improves with K Means initialization (SSIM: 0.73), a self-organized map VAE (SOMVAE), which maps each voxel onto a 2D grid is better able to preserve topological features of the image and achieve better reconstruction (SSIM: 0.82). We also test a recently invented probabilistic SOMVAE (PSOMVAE) which assigns a probability distribution over the whole grid for each voxel. In the end, while the PSOMVAE achieves best reconstruction performance (SSIM: 0.93) it results in an overly smooth SOM codebook that lies far from the encoder output. On the other hand, with SOMVAE, despite worse reconstruction, we were able to successfully embed closer brain regions onto closer nodes in the SOM grid.

Introduction:

One of the problems with traditional VAEs (variational autoencoders) with the domain of image reconstruction is that they result in images which can lack detail and sharpness. This is a result of the continuous multivariate normal $N(0, I)$ latent space used in traditional VAEs. To address this issue, van der Oord et al developed the VQVAE (vector quantized variational autoencoder) which consists of a categorical (discrete) latent space with dimension $e \in \mathbb{R}^{K \times D}$ where K is the number of embeddings and D is the dimension of each embedding vector [1]. Previously, in the context of MRI neuroimaging data, VQVAEs with 3D convolutions have been recently used to reconstruct T1 as well as diffusion weighted images (DWI) through a U-net (or U-net like) architecture [2]. However, one of the issues is that the codebook (mapping from the voxels to the embedding latent space) is a global codebook. Although better than a traditional VAE, it still does not capture local region information very well. To do so, it would be better to design a codebook that maps local brain regions to the same embeddings. Some work to preserve local information has been done in the context of 2D images, where a PixelCNN architecture was combined with the VQVAE to model a prior over the latent space. The goal of this work will be to perform a similar task of capturing local information for T1 weighted MRI data. To evaluate the models, we will use fidelity of reconstruction as measured by SSIM (structural similarity) with a uniform kernel and PSNR (peak signal to noise ratio). However, as the goal is also to ensure a rich codebook, we will assess the number of unique embeddings that are used by the quantizer layer. As such, a model with high SSIM and PSNR but a low number of unique embeddings is not useful for our purposes as the lack of embeddings indicates that no useful low-dimensional representations were being learned.

Dataset & Preprocessing:

We are using a neuroimaging dataset of 179 cognitively normal patients from the OASIS database. The dataset is split into 128 training, 23 validation, and 28 for testing. The images have 1 channel and are of shape (137,176,137) but were padded to become (144,176,144) for compatibility with

the base convolutional encoder-decoder architecture. Following this, the images were all Z-normalized to ensure the magnitude of the voxel values was not too large. A batch size of 32 was used for training.

Methods:

As a baseline, we started from the 3D VQVAE architecture from Ayub et al that was previously used to restore cropped DWI MRI images here https://github.com/RdoubleA/DWI-inpainting. The base architecture consists of a 3D convolutional encoder, vector quantizer codebook, and 3D deconvolutional decoder layer. The encoder consisted of 4 convolutions with batch norm and ReLU activations while the decoder consisted of 4 deconvolutions with batch norm and ReLU activations in all layers except for the output layer. The filter size (set to 4) doubled after each layer but was set equal to the embedding dimension in the last encoder layer before the codebook. In all the experiments shown, the dimension of the codebook was set to 32 while the number of embeddings was set to 256. The original model used for DWI MRI also made use of skip connections in a U Net architecture, although we found these to be suboptimal for learning a rich set of embeddings in the codebook upon experimentation, and thus they were turned off. A regular autoencoder (AE) with just the straight through encoder-decoder and a regular variational autoencoder (VAE) with sampling from *N(0, I)* in the latent space were also used as baselines to assess pure image reconstruction, so that we could determine how the inclusion of the codebook quantizer layer impacts this. Training for all models was performed with the Adam optimizer and learning rate of 0.005. The initial loss function used in the VQVAE was:

$$(1) \quad \mathcal{L}(x, E) = \left|\left|x - x_q'\right|\right|_2^2 + \sum_{j=1}^{J} \left(\alpha \left|\left| sg(z_e^j) - e_\tau^j \right|\right|_2^2 + \beta \left|\left| z_e^j - sg(e_\tau^j) \right|\right|_2^2 \right)$$

In the equation above, x_q' corresponds to the reconstructed image from the quantized representation (derived from the codebook), so the first term is the reconstruction loss. The other two terms respectively correspond to the codebook loss and commitment loss which act to ensure that the output of the encoder and the embeddings do not rapidly fluctuate [3]. It should be noted the original architecture for VQVAE only indirectly optimizes the codebook loss through an exponential moving average (EMA) update. Conceptually, during training the codebook weights (256 x 32) move closer to the 32 channel voxels of the incoming encoded image and vice versa. This can be seen as an online form of clustering, where each encoded voxel z_e^j is assigned to the nearest codebook weight e_τ^j and the network is updated via minibatch gradient descent. In addition to the loss function (which is not necessarily comparable between different codebook architectures), the structural similarity (SSIM) and peak-signal-to-noise ratio (PSNR) are used to evaluate the reconstruction. Additionally, to check that the codebook is being updated correctly, we performed t-distributed stochastic neighbor embedding (TSNE) with the encoder output and codebook as well as assessed the number of unique embeddings on a selected image in the training set.

One of the problems we encountered with the baseline VQVAE model was that of the codebook initialization. The first thing we tried is using the regular AE weights as initializations for the encoder and decoder, and exploring a standard K means clustering based initialization based on the regular AE encoder output. Furthermore, the original VQVAE loss in equation (1) only uses the reconstruction output from the quantized representation of the encoded image. As this could result in worse reconstruction due to too much compression, we opted to try a model which added a reconstruction term based on the reconstruction from the direct encoder output x_e' (Note: $\gamma_e + \gamma_q = 1$, we chose $(\gamma_e, \gamma_q) = (0.8, 0.2)$). This idea has been used in a similar self- organized-map (SOM) codebook architecture [4]

$$(2) \quad \mathcal{L}(x, E) = \gamma_e \left|\left|x - x_e'\right|\right|_2^2 + \gamma_q \left|\left|x - x_q'\right|\right|_2^2 + \sum_{j=1}^{J} \left(\alpha \left|\left| sg(z_e^j) - e_\tau^j \right|\right|_2^2 + \beta \left|\left| z_e^j - sg(e_\tau^j) \right|\right|_2^2 \right)$$

In addition to the VQ codebook architecture we also explored the actual self-organized map (SOM) based codebook grid, also with a K means initialization [4]. Unlike the VQ codebook which

consists of disjoint clusters, the idea of a SOM is to enforce more of a topological structure by organizing its nodes (which each consist of a 32-dimensional embedding vector) in a grid. At a high level, each voxel is still mapped to a best matching unit (BMU) vector in a node like the VQ, but this time the immediate neighborhood of nodes 1 away to the BMU also gets updated to be closer to the BMU. Both a rectangular grid (where all edge nodes would have fewer neighbors) and a toroidal grid (nodes at the border of the corresponding rectangular grid have a neighbor on the opposite side, thus all nodes have an equal neighborhood size) were explored. The codebook loss (the last 2 terms in (2)) were modified to include a term to keep the quantized and encoded representation similar as well as a term to keep the neighborhood of the BMU (including itself) close to the encoder output [4]:

$$(3) \quad \mathcal{L}(x,E) = \gamma_e \left\|x - x'_e\right\|_2^2 + \gamma_q \left\|x - x'_q\right\|_2^2 + \sum_{j=1}^{J}\left(\alpha \left\|z_{e,j} - z_{q,j}\right\|_2^2 + \beta \sum_{N(z_q)} \left\|\tilde{e}_j - sg(z_{e,j})\right\|_2^2\right)$$

Our SOM grid also used K means for initialization of the codebook, but we also tried the original Kohonen non-gradient based SOM as initialization as done by some literature since we postulated this could make the nodes closer together to start with and lead to faster convergence of the codebook loss [5]. The SOM model in Equation (3) still uses hard cluster (node) assignment. As hard cluster assignment could result in loss of information content in the image, another model we explored was a probabilistic SOM introduced in Manduchi et al, which assigns probabilities s_{ij} to each node j for a given voxel i based on the t-distribution with degrees of freedom df. In this model, there is no quantized representation and the direct encoder output is fed to the decoder, but loss terms corresponding to the SOM codebook are added.

$$(3) \quad s_{ij} = \frac{\left(1 + \left\|z_i - \mu_j\right\|/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'}\left(1 + \left\|z_i - \mu_j\right\|/\alpha\right)^{-\frac{\alpha+1}{2}}}$$

$$(4) \quad t_{ij} = \frac{s_{ij}^2 / \sum_{i'} s_{i'j}}{\sum_{j'}\left(s_{ij}^2 / \sum_{i'} s_{i'j'}\right)}$$

$$(5) \quad \mathcal{L}(x,E) = \left\|x - x'_e\right\|_2^2 + \gamma \sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij} \log \frac{t_{ij}}{s_{ij}} - \beta \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} s_{ij} \sum_{e \in N(j)} \log s_{ie}$$

In equation (5), increasing γ results in a greater degree of hard cluster assignment, while increasing β results in a smoother SOM grid, with neighbors closer to each other [5]. Note that the PSOM VAE inherently is forced to use all the embeddings in the codebook (to varying degrees). For the SOM-VAE and PSOM-VAE we also experimented with using a sampled encoded output from a N(0, I) latent space layer along with a standard VAE KL divergence loss instead of using the encoded output directly. We call these models the "continuous" VQVAE, SOM-VAE, PSOM-VAE referring to the continuous N(0,I) latent space.

Results:

Model	Init	Epochs	Loss Eqn	Hyperparameters	Train MSE (Val MSE)	Train SSIM (Val SSIM)	Train PSNR (Val PSNR) (dB)	# of unique embeddings used
Base VQVAE w/Skip (rand init)	rand	100	(1)	$\beta = 6$	0.00025 (0.00035)	0.97 (0.97)	33.1 (33.7)	3/256
Base VQVAE (rand init)	rand	120	(1)	$\beta = 6$	0.0045 (0.0062)	0.60 (0.61)	19.5 (20.3)	3/256
Regular AE	---	240	---	---	0.0011 (0.0016)	0.87 (0.87)	24.9 (25.5)	---
Regular VAE	--	250	---	---	0.0160 (0.022)	0.57 (0.58)	19.3 (20.1)	---
VQVAE w/Skip	AE KM	100	(1)	$\beta = 6$		0.98 (0.98)	32.8 (33.5)	1/256
VQVAE	AE KM	120	(1)	$\beta = 6$	0.0060 (0.0084)	0.63 (0.63)	19.8 (20.6)	46/256
VQVAE (AE KM, rand wts)	AE KM rand wt	120	(1)	$\beta = 6$	0.0063 (0.0087)	0.50 (0.52)	18.3 (19.0)	2/256
New VQVAE	AE KM	120	(2)	$\beta = 6, \gamma_e = 0.5, \gamma_q = 0.5$	0.0055 (0.0067)	0.72 (0.72)	21.7 (22.4)	58/256
New VQVAE	AE KM	120	(2)	$\beta = 6, \gamma_e = 0.8, \gamma_q = 0.2$	0.0045 (0.0063)	0.73 (0.73)	21.7 (22.3)	56/256

SOMVAE rectangular	AE KM	700	(3)	$\alpha = 6, \beta = 1, \gamma_e = 0.8, \gamma_q = 0.2$	0.30 (0.45)	0.82 (0.82)	23.5 (24.1)	113/256
SOMVAE rectangular	AE SOM	700	(3)	$\alpha = 6, \beta = 1, \gamma_e = 0.8, \gamma_q = 0.2$	0.39 (0.74)	0.81 (0.81)	23.3 (24.0)	51/256
SOMVAE toroid	AE KM	700	(3)	$\alpha = 6, \beta = 1, \gamma_e = 0.8, \gamma_q = 0.2$	0.17 (0.23)	0.82 (0.82)	23.6 (24.2)	90/256
SOMVAE toroid Continuous latent	VAE KM	700	(3) + VAE KL div	$\alpha = 6, \beta = 1, \gamma_e = 0.8, \gamma_q = 0.2$	0.986 (1.37)	0.62 (0.63)	19.9 (20.6)	6/256
PSOMVAE toroid	**AE wts only**	**800**	**(4)**	$\boldsymbol{\gamma = 1, \beta = 1}$	**0.71 (0.98)**	**0.91 (0.91)**	**26.4 (27.0)**	**---**
PSOMVAE toroid	**AE wts only**	**800**	**(4)**	$\boldsymbol{\gamma = 1, \beta = 0.3}$	**0.21 (0.29)**	**0.91 (0.91)**	**26.4 (27.0)**	**---**
PSOMVAE toroid	AE KM	800	(4)	$\gamma = 10, \beta = 1$	1.05 (1.34)	0.90 (0.90)	25.9 (26.5)	---
PSOMVAE toroid Continuous latent	VAE wts only	800	(4) + VAE KL div	$\gamma = 1, \beta = 1$	1.88 (2.56)	0.62 (0.62)	19.9 (20.6)	---

Table 1: Experimental Results. The PSOMVAE performs the best (outside of the baseline VQVAE, which didn't learn any embeddings) in terms of reconstruction SSIM and PSNR. Note the MSE loss values are not comparable across different models. Other codebook architectures perform worse than a regular AE. The continuous VAE based architectures (and a regular VAE) perform surprisingly poorly for this dataset. No practical overfitting is suspected of any model as the validation SSIM and PSNR metrics are very close to the training, even if the loss may be slightly higher. Choice of hyperparameters for the PSOMVAE does not seem to impact the results substantially.

According to Table 1, the baseline VQVAE with skip connections is able to perform the best on the pure image reconstruction task with SSIM=0.97, but fails to accomplish the goal of learning a rich codebook as evidenced by only 3/256 embeddings used. We suspect the skip connections offered the model a way to easily cheat and simply copy the input to the output. Indeed, when the skip connections were turned off, the baseline VQVAE model reconstruction dropped to SSIM=0.60 though the model still learned no substantial embeddings. This is also evidenced by the TSNE plot in Figure A1, where the dictionary (codebook) points are not mixed in well with the encoder output. We suspected this could be caused by poor codebook initialization, which has not been discussed substantially as previous work simply used a random normal or uniform initialization [1, 2]. Using K means initialization marginally improved the reconstruction SSIM to 0.63, but the main improvement was in the # of unique embeddings, which became 46/256. Interestingly, K Means initialization was not enough and the model required weight initialization based on the regular AE (which had SSIM = 0.87). The fact that the baseline VQVAE with skip connections also failed to learn substantial embeddings even after AE weight and K means initialization confirmed our intuition that skip connections are a poor choice for learning a rich codebook. The actual number of unique embeddings used was an issue which was not explored in the original paper for the baseline model by Ayub et al. At this point, our main issue was still that the codebook was not being fully utilized, and the reconstruction SSIM was still much worse than a regular AE. By adding a loss term based on the direct encoder output, we were able to improve the # of unique embeddings to 53/256 and SSIM to 0.73. This is likely a result of allowing gradient information of the reconstruction loss to pass to the codebook further, whereas this does not happen with just a loss based on the quantized reconstruction [4].

The biggest improvement came when we decided to change the architecture entirely to that of a SOM grid. This was motivated by the intuition that the VQVAE embeddings were very disjoint, and there may be loss of topological information. Only a subset of embedddings were actually being used, so we intuitively felt that if other embeddings could be forced closer to this subset, they may also become occupied. This was confirmed when we saw the reconstruction SSIM improve to 0.82 and number of unique embeddings used to 105/256 with a rectangular SOM grid. A toroidal grid had the same reconstruction SSIM but a slightly lower number of unique embeddings at 90/256 used. A K means initialization for the SOM VAE performed better than a regular SOM (non-gradient based) initialization, despite the latter having been used in other work. The TSNE for the SOMVAE in Figure 1 visually confirms the encoder output lies close to the codebook. In the case of the SOMVAE, we can

approximately visualize the topology of the grid by summing across the channel dimension of 32 in Figure 1. Quantitatively, the Pearson correlations between neighbors were also found to be >0.95 in most cases. However, since the reconstruction SSIM was still below that of the regular AE and the full embeddings were not being utilized, we investigated the probabilistic SOM (PSOM). The PSOM essentially forces the model to use all embeddings and nodes for every voxel, to varying degrees based on a probability rather than hard cluster assignment. The PSOM VAE with a toroidal grid resulted in an SSIM of 0.91 and was not very sensitive to the hyperparameters of the loss. The map (summed across the channel dimension) is shown in Figure 2. For the SOMVAE, based on the brain image slice in Figure 2, we appear to have successfully mapped local brain regions onto the same locations in the grid, thus preserving topology. In the end, we confirmed the lack of overfitting by evaluating the SOMVAE and PSOMVAE on the test set, where the SSIM=0.81 and SSIM = 0.90 respectively.

For the PSOMVAE with almost all the nodes surprisingly having a very similar vector. In theory, we suspected the over smoothness of the grid would be due to the hyperparameters of the PSOMVAE but these did not empirically affect the picture much [5]. To investigate this further, we performed TSNE and as it would turn out—the image encodings and dictionary were unfortunately very separated. This may be due to the lack of a quantized reconstruction term in the loss in the PSOMVAE equation (4) as opposed to the SOMVAE equation (3). The TSNE plot suggests the model is likely finding a region around a single point for all the codebook weights. Thus, although the reconstruction of the PSOMVAE as of now is very good, the model is not finding a codebook which captures the information in the image. The SOMVAE does this better but suffers from worse reconstruction SSIM compared to the standard AE. Interestingly, we found in experiments when K Means initialization was used for the PSOMVAE, the loss diverged within the first 10 epochs and training was interrupted. This observation combined with Figure 1 suggests the PSOMVAE prefers to keep the codebook and encoder output well separated, though we don't know why nor why this may result in improved reconstruction SSIM relative to the regular AE. Lastly, we also tested a continuous latent space VAE along with the SOMVAE and PSOMVAE algorithms, but based on Table 1 this performed worse than using a regular AE in the encoder, which is in contrast to the findings of Manduchi et al [5].

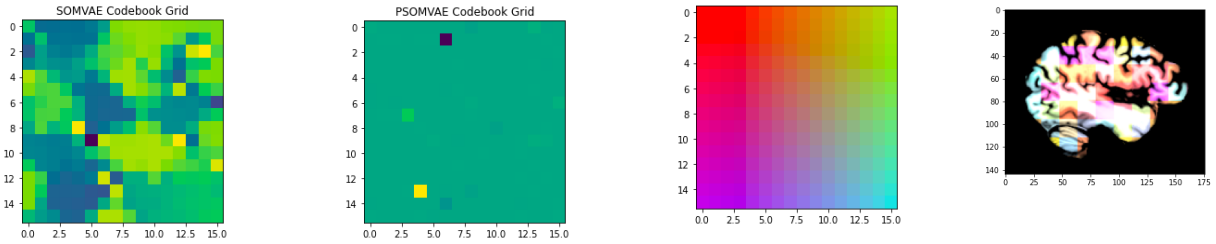

Figure 1: SOM and PSOM codebooks and the projection of the SOM grid onto a brain slice. Similar and closer together regions of the brain are mapped onto closer regions on the grid, which the colors show.

Conclusion/Future Work:

Overall, in this work we tested many different architectures, hyperparameters, and initializations for the codebook. It was found that codebook initialization via K Means on a pre-trained regular AE is crucial to successfully learning a substantial number of embeddings. Additionally, we found that a SOM grid-based codebook helps preserve the topology of the input brain image: brain regions closer together are assigned to the same or neighbor nodes which can be visualized in Figure 1. Using a PSOM codebook improved the reconstruction but resulted in a very homogenous smooth map which was far from the encoder output. In the future, we would like to explore how the PSOM can be modified to achieve a similar grid-based codebook as the SOM while simultaneously retaining reconstruction fidelity as assessed by SSIM.

Contributions:

My mentors Dr. Zhao and PhD student Jiahong Ouyang helped guide me with initial ideas/suggestions for the experiments such as the K means initialization and considering using the direct encoder output reconstruction as an additional term in the VQVAE loss. They also suggested and explained the idea of TSNE to me as an additional check beyond the # of unique embeddings to make sure the model was learning properly. Beyond the base VQVAE model from this github: https://github.com/RdoubleA/DWI-inpainting all implementation/ modification of the ideas was done by me. I came up with the idea of trying the SOM-VAE and PSOM-VAE after reading the corresponding papers and then implementing the algorithms for my dataset.

References:

1. van den Oord, A., O. Vinyals, and K. Kavukcuoglu *Neural Discrete Representation Learning*. 2017. arXiv:1711.00937.
2. Ayub, R., et al. *Inpainting Cropped Diffusion MRI Using Deep Generative Models*. 2020. Cham: Springer International Publishing.
3. Tudosiu, P.-D., et al. *Neuromorphologicaly-preserving Volumetric data encoding using VQ-VAE*. 2020. arXiv:2002.05692.
4. Fortuin, V., et al. *SOM-VAE: Interpretable Discrete Representation Learning on Time Series*. 2018. arXiv:1806.02199.
5. Manduchi, L., et al., *T-DPSOM: an interpretable clustering method for unsupervised learning of patient health states*, in *Proceedings of the Conference on Health, Inference, and Learning*. 2021, Association for Computing Machinery. p. 236–245.

Appendix:

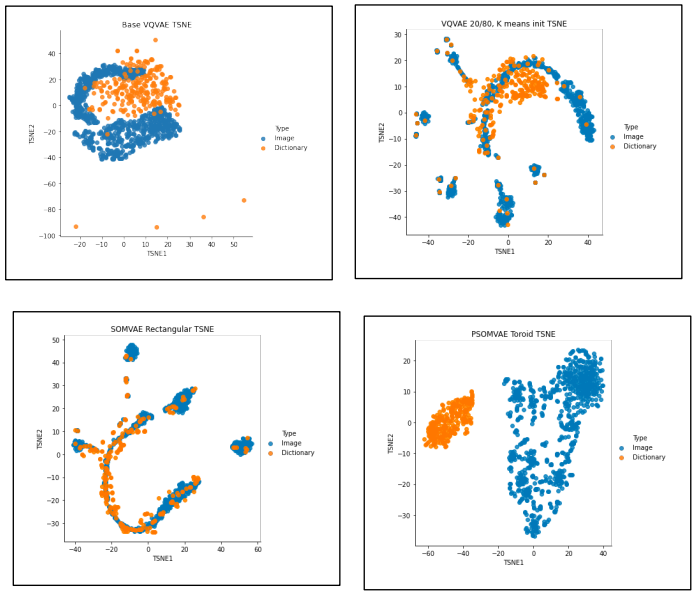

Figure A1: TSNE of codebook/dictionary and encoder output from a sample image in the training set. The baseline VQVAE (no skip connections) has a border between the encoder output and dictionary. The encoder output and dictionary are more well mixed in the VQVAE with K means initialization and very well mixed with the SOMVAE. For the PSOMVAE, the encoder output and codebook are very distinct.

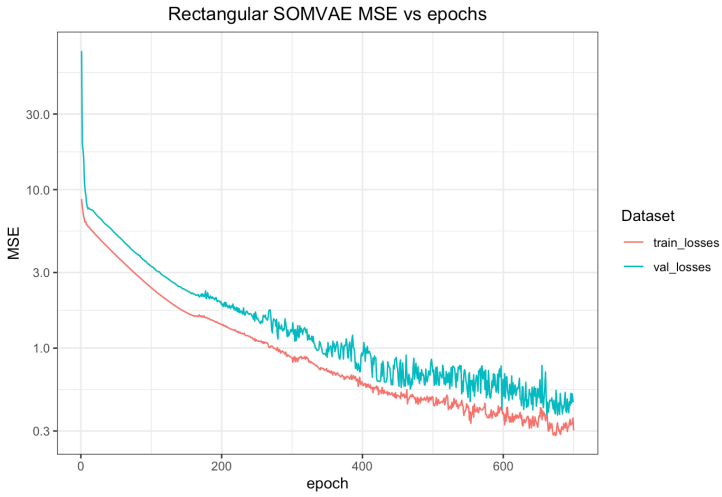

Figure A2: MSE vs epochs loss curves for training and validation data for the SOMVAE. Training becomes noisier toward the end.