Assignment Two: Inference

Anjie Song
1705056
as17472

Xiang He
1723303
xh17500

Xing Li
1727134
xl17507

Names not listed in order

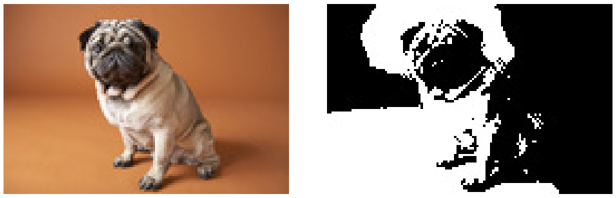


Figure 1: The original colored and grey scaled images

Question1

In this report, we are all using 8 neighbors. The prior and likelihood functions for the x_i used in ICM is shown below,

$$p(x_i) = \frac{1}{z_0} e^{E(x_i)}$$

$$p(y \mid x) = \frac{1}{z_1} e^{L(x_i, y_i)}$$

where

$$E(x_i) = \beta \times \sum_{j=neighbors(x)} x_i x_j$$

$$L(x_i, y_i) = \eta \times x_i y_i$$

and the β is set to 1.5 and η is 2.0.
Figure 2 shows the ICM results with different iteration number. It is obvious that the result has been convergence from iteration 3. Figure 3 demonstrate the ICM results with different noise level. The performance of denoising is terrible when the probability is under 0.8.
Though the ICM technology can perform denoising, it does not have ability to recover some details of the original image.

Question 2

In the Gibbs sampling, the prior and likelihood functions are the same as the ICM ones except that $\beta = 0.8$ and $\eta = 5$. Figure 4 gives the results in different noise level and the iteration number is fixed to 5. We can see that the effect of denoising starts to drop down from variance is 0.4. Since we set η to high value, the posterior has the strong association with y value which means the result tends to imitate the original value unless the prior has strong evidence to change its mind.

Question 3

The upside plots in Figure 5 shows the results without random index and the downside plots is with random index. There is no different between these two approach since the Gibbs sampling only updates the nodes of the next iteration and still computes the posterior of the nodes in the current iteration, therefore, the order of the iteration is no matter.

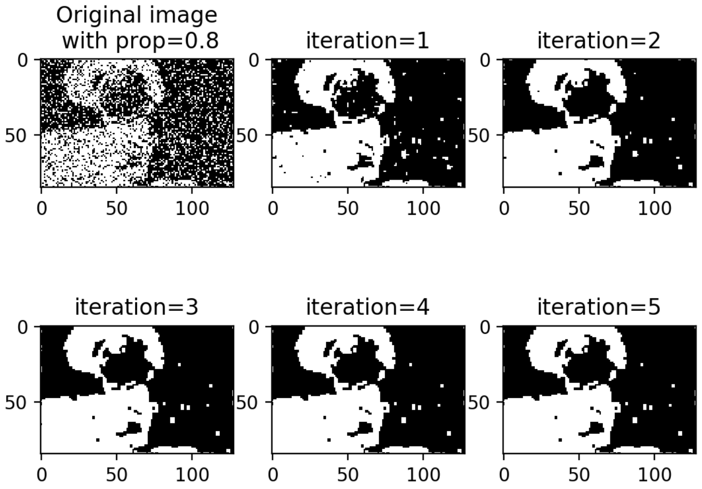


Figure 2: ICM with different iteration number

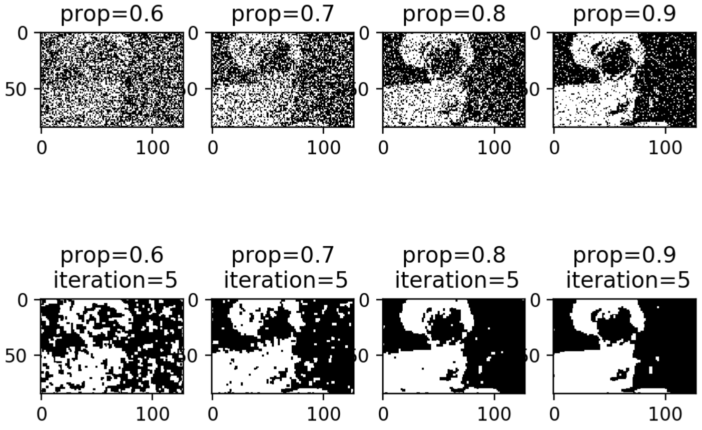


Figure 3: ICM with different noise level

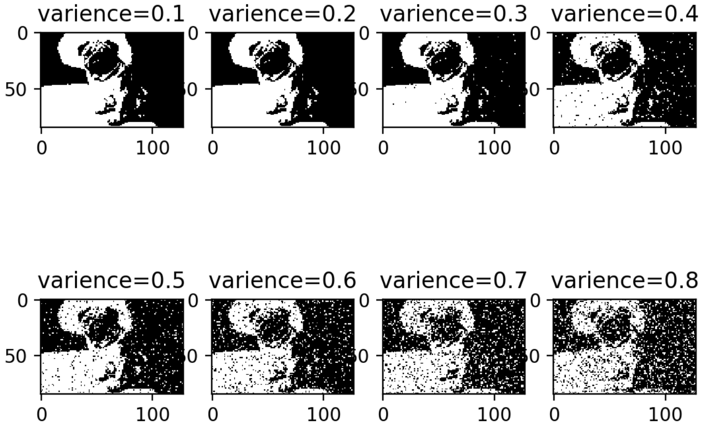


Figure 4: Gibbs denoise with different noise level

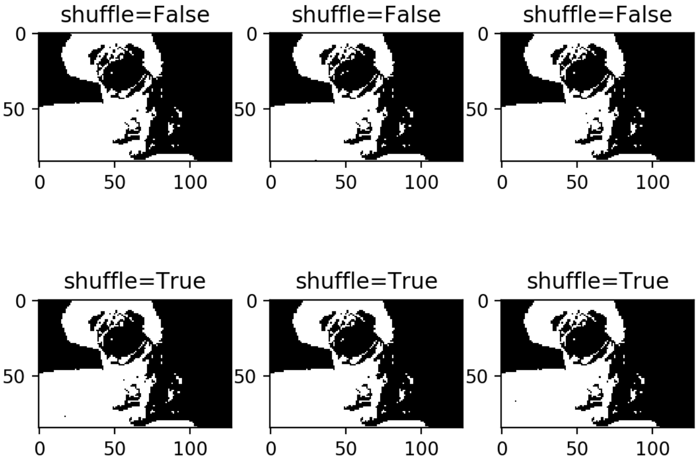


Figure 5: Gibbs denoise with randomly picking or not

Question 4

The result shows the result is convergence from iteration 1 and there is no significant distinguish in the later iterations.

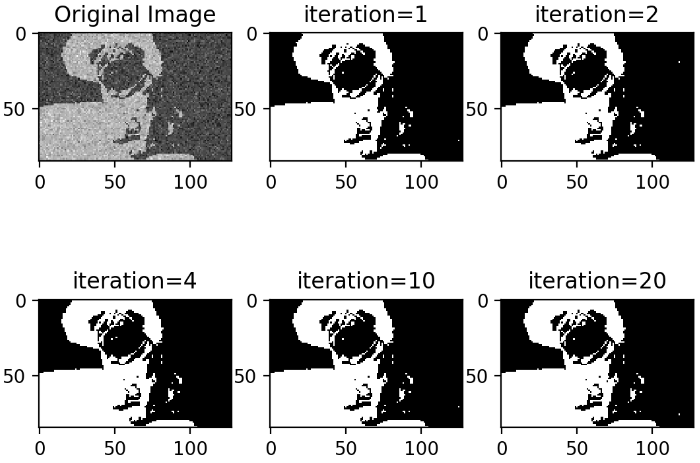
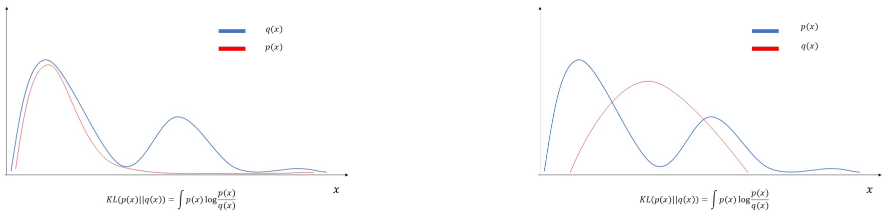


Figure 6: Gibbs denoise with different iteration number

Question 5

The ideal approximate distribution $q(x)$ is the one which is as same as $p(x)$, and the KL divergence means the difference between two distributions. For the $KL(p \mid\mid q)$, or forward KL, if $p(x) = 0$, the approximate distribution $q(x)$ can be random value since the equation $p(x)log\frac{p(x)}{q(x)}$ will still be close to 0 no matter what the $q(x)$ is. And when $q(x) = 0$, if $p(x) > 0$, the KL divergence can be very big, thus the optimization will try to minimize it, which is shown in Figure 7. That is why forward KL is known as **zero avoiding**. And for the $KL(q \mid\mid p)$, or the reverse KL, if $q(x)$ is close to 0, the approximate distribution $p(x) > 0$ will not impose a penalty to the divergence. If $q(x) > 0$, the difference between $q(x)$ and $p(x)$ must be as low as possible, which contribute to the overall divergence.



(a) Before optimization

(b) After optimization

Figure 7: Problem of Forward KL divergence

Question 6

The denoising results of mean field variational Bayes in Ising model is shown in Figure 8. When using this algorithm, we choose the L function as

$$L(i) = x_i * y_i * weight$$

with $weight$ equals 5 and the w_{ij} in prior are all set to 0.1. The max iteration is set to 5.

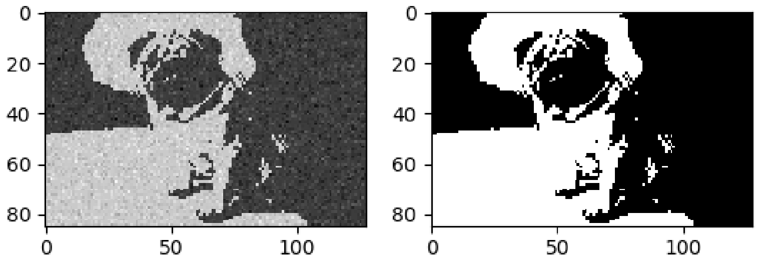
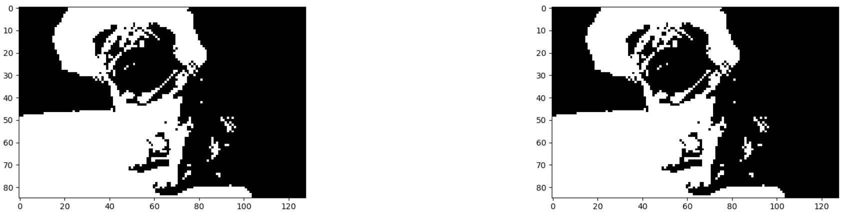


Figure 8: Mean field variational Bayes

Question 7

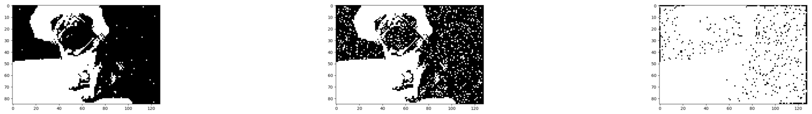
When we change the time of iteration, we can find that for the first time of iteration, variational Bayes can get a very good performance(may cause overfitting?). And when we change the parameters of prior w_{ij}, the iteration time to get a good performance will vary. However, it really can get a better performance than the sampling methods since it is the approximation of the real posterior distribution. And the iteration time we need is also smaller. It has the advantage of maximizing an explicit objective.



(a) iteration = 1

(b) iteration = 3

Figure 9: Change the iteration time



(a) $w_{ij} = 0.3$ (b) $w_{ij} = 0.5$ (c) $w_{ij} = 0.8$

Figure 10: Change the parameters with iteration = 1

Question 8

We tried mean field variational Bayes to implement the image segmentation. We used the histograms as the evaluation of likelihood function. It can get a good performance when the image has a very clear difference of colour between the foreground and background. That is, the histograms of the whole image has two clear peaks. Then we choose a threshold to separate them to two histograms, one is background and the other is foreground. With this new likelihood, we can perform the mean field image segmentation. The Figure 13 shows the normalized histogram of image we used. And when some region of the background has similar colour with the foreground, it will be given a foreground label for a large chance(it is actually has a big relation with colour), as shown in Figure 12b. It will be a disadvantage. In Figure 14 and 15 we used 5 × 5 and 25 × 25 Gaussian smooth to the histograms respectively and achieved better performance obviously(still not perfect). And then, we choose another image which the histogram has only one peak, the performance then is very bad, which is shown in Figure 16.

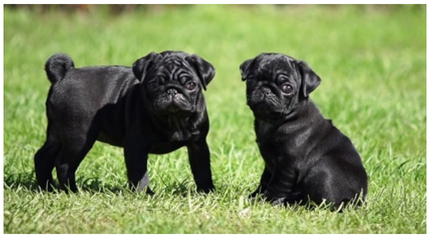
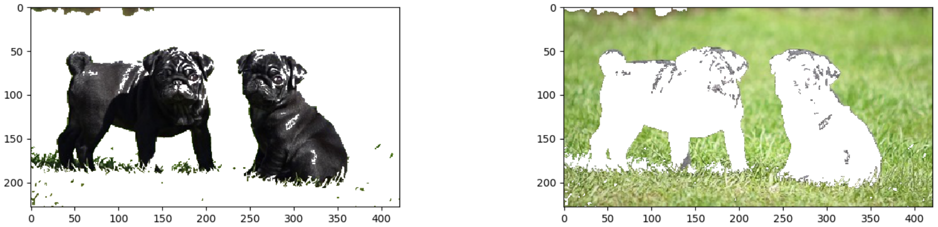


Figure 11: Original image



(a) Foreground (b) Background

Figure 12: Mean field image segmentation

Question 9

1) This report uses the original version's VAE provided by [1] to produce the diagram of the architecture of VAE17. VAE could be divided into two parts. The first part is called Encoder, in terms of MNIST dataset, it takes one image as input and then uses neural networks to learn the parameters of the distribution of latent variables. Here there is a strong assumption that we assume the distribution of latent variables are some known distribution so that we could handle this distribution by

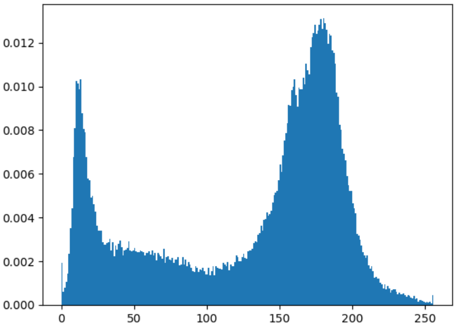
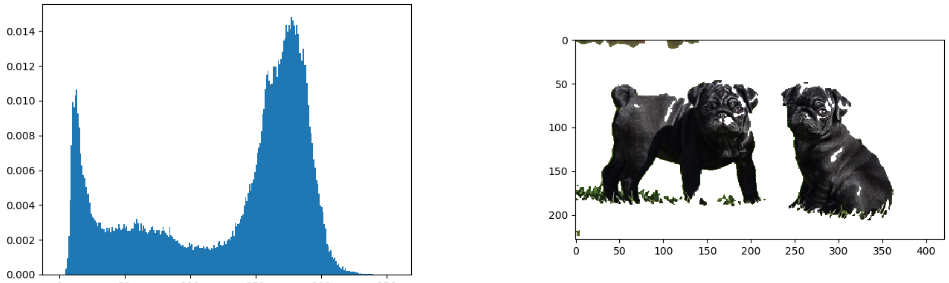


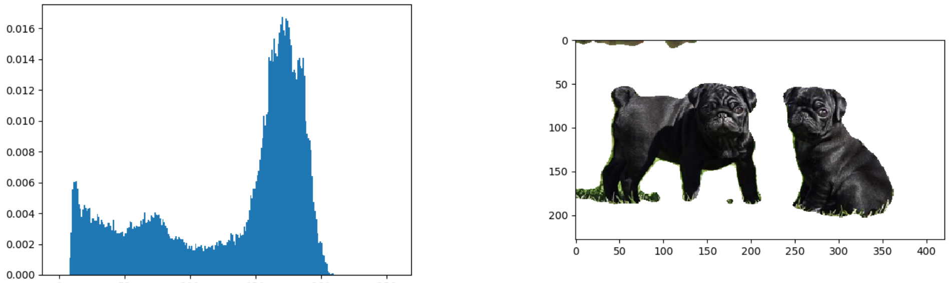
Figure 13: Normalized histogram



(a) Histogram

(b) Foreground

Figure 14: Mean field image segmentation with 5X5 Gaussian smooth



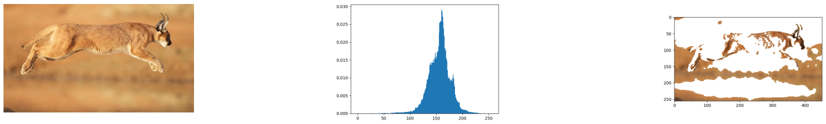
(a) Histogram

(b) Foreground

Figure 15: Mean field image segmentation with 25X25 Gaussian smooth

learning its parameters through our encoder neural network. Following the learnt latent variables' distribution, we sample a point from this distribution and put it into another reconstruction neural network, which is called decoder, to generate an image. VAE uses KL distance loss to force the latent variables' distribution into approximating the normal distribution and some measurement of loss, like cross entropy or mean square error, between the input image and output image to force the decoder network into learning the mapping from latent variables space to data space.

An important trick used in VAE is reparameterisation. One pitfall hidden in above description is that the operation we sample from the latent variables space is not a common one like addition,



(a) Original image

(b) Histogram

(c) Foreground

Figure 16: Image segmentation for one peak histogram image

multiplication or division. However, backpropagation is widely used in the modern neural network to carry the task of gradient descent and this method is powerless facing the sampling operation, in other words, the whole network is interrupted by sampling. This trick produces an ϵ which is sampled from a standard normal distribution and then use this ϵ combined with learnt parameters to imitate the sampling process. After applying this trick, the whole network now could be trained.

2)To explain the difference between Variational Auto-Encoder and Variational Bayes, the graphic model is an excellent illustration18. In one word, VB is a method of approximation to approximate some intractable posterior $q_\phi(Z|X)$ where we have to give assumption on our prior distribution and likelihood and generally the result is discrete. On the other hand, VAE tries to learn the parameters ϕ and θ in both of the posterior $q_\phi(Z|X)$ and the likelihood $p_\theta(X|Z)$ where we usually assume the latent variable is subject to Gaussian distribution.

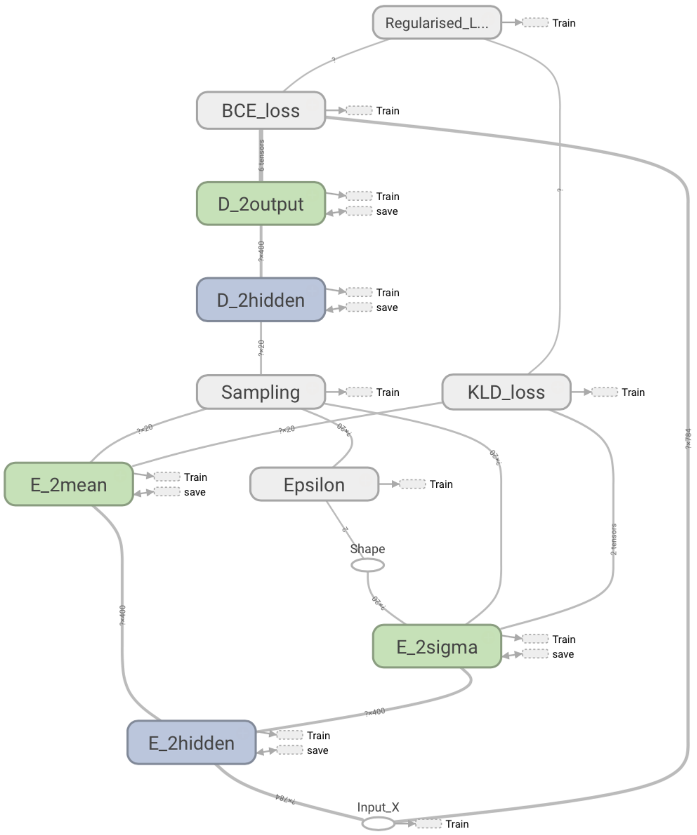


Figure 17: The brief architecture of Variational Auto-Encoder

Question 10

Based on question 9, the key point is to explore the distribution of latent variables **Z** and the mapping function or likelihood $p_\theta(X|Z)$ where X means the data space. While the network of decoder

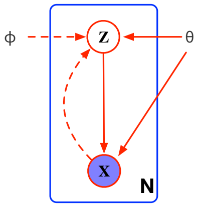


Figure 18: The graphic model of VAE and VB

has numerous parameters, it is intractable to explain the meaning of each of them. For the convenience of visualisation, we assume the latent variable space is two dimensions, and we uniformly sample twenty points in each dimension. This sampling repeats three times for three different scale separately:$[-1,1]$19, $[-2,2]$20 and $[-5,5]$21.

From these three visualisations, it clearly shows the mapping from latent variables space to data space, and according to "68−95−99.7 rule"[1], scale[-2,2] contains 95% values. We could see "1" and "0" are simple to be distinguished and generated while some characters have confusing shape are gathering in the centre like "3","8","9" or "4". (Merry Christmas!)

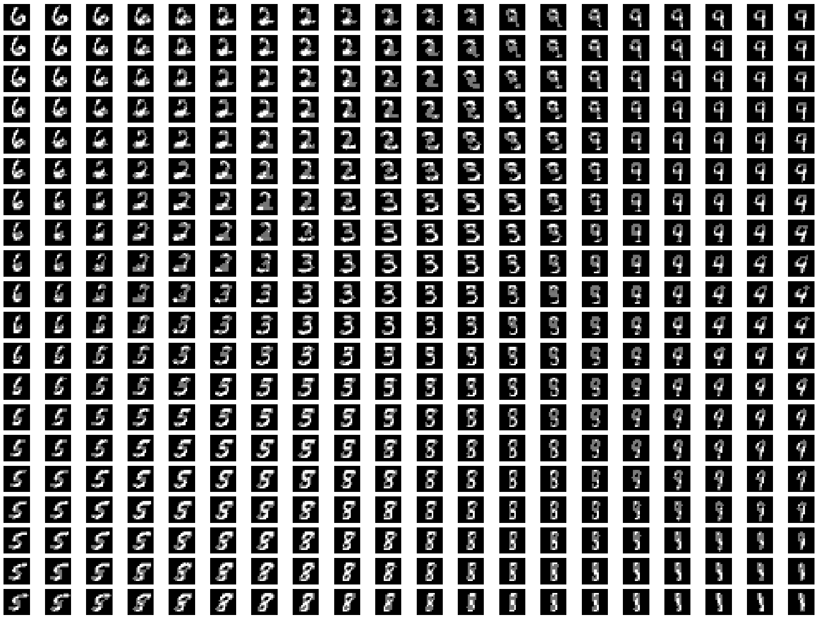


Figure 19: scale$[-1,1]$

References

[1] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[1] https://en.wikipedia.org/wiki/68-95-99.7_rule

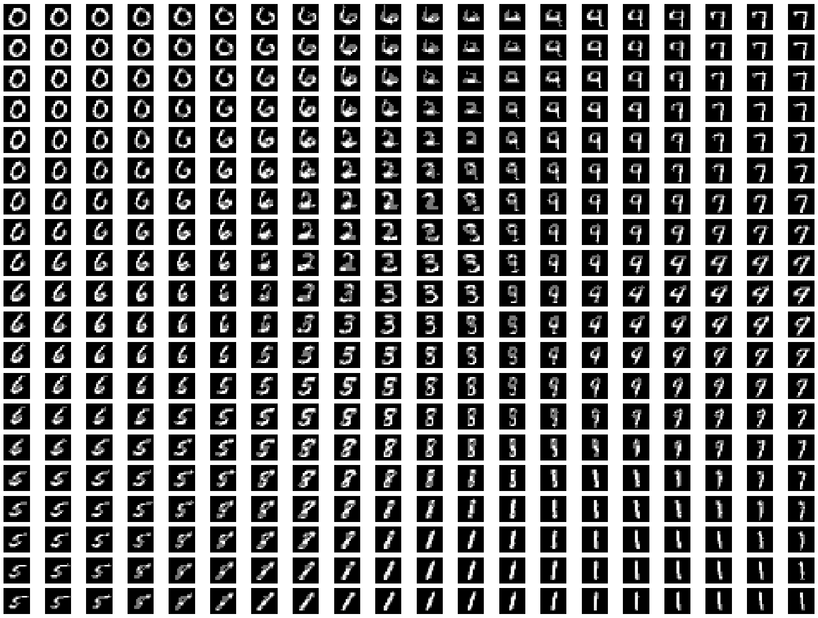


Figure 20: scale$[-2, 2]$

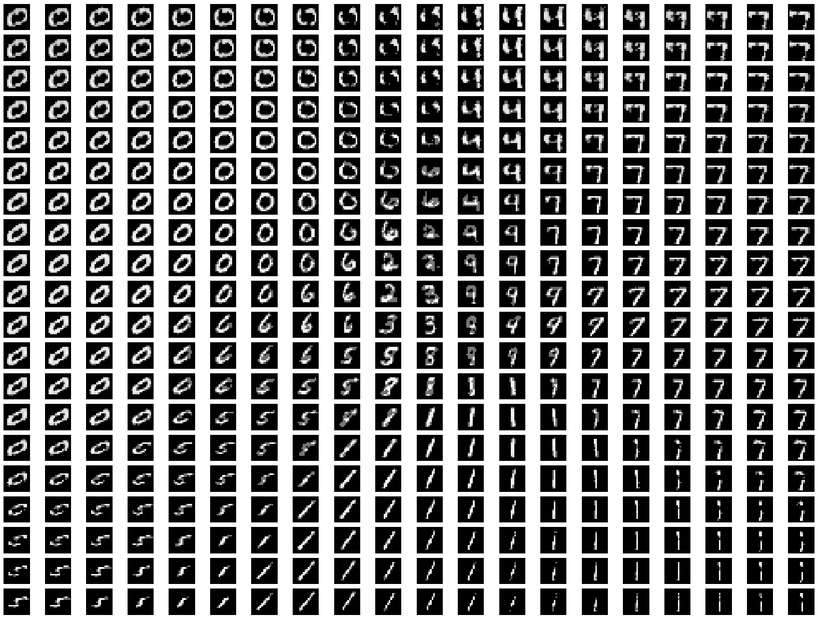


Figure 21: scale$[-5, 5]$