Homework 04

Deep Learning

Name: Raja Haseeb ur Rehman

Student ID: 20194673

Problem 1

- The original objective function is $\min_{\phi,\theta} D_{KL}(q_\phi(z|x)||p_\theta(z|x))$. Derive $\text{argmin}_{\phi,\theta} D_{KL}(q_\phi(z|x)||p_\theta(z|x)) = \text{argmax}_{\phi,\theta}\mathcal{L}(\theta,\phi;\mathbf{x}^{(i)})$ (3 points)
- For constrained optimization problem, $\min_{\phi,\theta} D_{KL}(q_\phi(z|x)||p_\theta(z|x))$ s.t. $D_{KL}(q_\phi(z|x)||p(z)) < \epsilon$ can be approximated to $\max_{\phi,\theta} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \lambda D_{KL}(q_\phi(z|x)||p(z))$. Derive the formula by using Lagrange multiplier method. Note that ϵ is positive real number. (You can use above result) (3 points)

Solution

Notably, we have some liberty to choose some structure for our latent variables. We can obtain a closed form for the loss function if we choose a Gaussian representation for the latent prior *p*(z) and the approximate posterior, *qθ*(z | *xi*). In addition to yielding a closed form loss function, the Gaussian model enforces a form of regularization in which the approximate posterior have variation or spread (like a Gaussian).

Closed form VAE Loss: Gaussian Latents

Say we chose:

$$p(z) \to \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right)$$

and

$$q_\theta(z|x_i) \to \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)$$

then the KL or regularization term in the ELBO becomes:

$$-D_{KL}(q_\theta(z|x_i)||p(z)) =$$

$$\int \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}\right) dz$$

After evaluating the term in logarithm and simplifying, we get

$$\frac{1}{\sqrt{2\pi\sigma_q^2}} \int \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right) \left\{\log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right\} dz.$$

Expressing the above as expectation we get

$$
\begin{aligned}
-D_{KL}(q_\theta(z|x_i)||p(z)) &= E_q\left\{\log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right\} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) + E_q\left\{-\frac{(x-\mu_p)^2}{2\sigma_p^2} + \frac{(x-\mu_q)^2}{2\sigma_q^2}\right\} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left\{(x-\mu_p)^2\right\} + \frac{1}{2\sigma_q^2}E_q\left\{(x-\mu_q)^2\right\}
\end{aligned}
$$

Moreover, since the variance σ2 is the expectation of the squared distance from the mean, i.e.

$$\sigma_q^2 = E_q\left\{(x-\mu_q)^2\right\},$$

It follows that,

$$
\begin{aligned}
-D_{KL}(q_\theta(z|x_i)||p(z)) &= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left\{(x-\mu_p)^2\right\} + \frac{\sigma_q^2}{2\sigma_q^2} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left\{(x-\mu_p)^2\right\} + \frac{1}{2} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left\{(x-\mu_q+\mu_q-\mu_p)^2\right\} + \frac{1}{2} \\
&= \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left\{(\underbrace{x-\mu_q}_{a} + \underbrace{\mu_q-\mu_p}_{b})^2\right\} + \frac{1}{2}
\end{aligned}
$$

And when we take $\sigma_p = 1$ and $\mu_p = 0$, we get,

$$
\begin{aligned}
-D_{KL}(q_\theta(z|x_i)||p(z)) &= \log(\sigma_q) - \frac{\sigma_q^2+\mu_q^2}{2} + \frac{1}{2} \\
&= \frac{1}{2}\log\left(\sigma_q^2\right) - \frac{\sigma_q^2+\mu_q^2}{2} + \frac{1}{2} \\
&= \frac{1}{2}\left[1 + \log\left(\sigma_q^2\right) - \sigma_q^2 - \mu_q^2\right]
\end{aligned}
$$

Recall the ELBO,

$$\log p(x_i) \geq -D_{KL}(q_\theta(z|x_i)||p(z)) + E_{\sim q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]$$

From which it follows that the contribution from a given datum xi and a single stochastic draw towards the objective to be maximized is,

$$\frac{1}{2}\left[1+\log\left(\sigma_j^2\right)-\sigma_j^2-\mu_j^2\right]+E_{\sim q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right]$$

where σ_j^2 and μ_j are parameters into the approximate distribution, q, and j is an index into the latent vector z. For a batch, the objective function is therefore given by,

$$\mathcal{G}=\sum_{j=1}^{J}\frac{1}{2}\left[1+\log\left(\sigma_i^2\right)-\sigma_i^2-\mu_i^2\right]+\frac{1}{L}\sum_l E_{\sim q_\theta(z|x_i)}\left[\log p(x_i|z^{(i,l)})\right]$$

To obtain the loss function, we simply take the negative of G:

$$\mathcal{L}=-\sum_{j=1}^{J}\frac{1}{2}\left[1+\log\left(\sigma_i^2\right)-\sigma_i^2-\mu_i^2\right]-\frac{1}{L}\sum_l E_{\sim q_\theta(z|x_i)}\left[\log p(x_i|z^{(i,l)})\right]$$

Therefore, to train the VAE is to seek the optimal network parameters ($\theta*$; $\varphi*$) that minimize L:

$$(\theta^*,\phi^*)=argmin_{(\theta,\phi)}\mathcal{L}(\theta,\phi)$$

Problem 2

- For original VAE ($\lambda = 1$), perform a qualitative comparison of samples according to the size of the hidden dimension and describe the results. ($H = 2; 10; 25; 50$)
- What can be strength and weakness for VAE with large hidden dimension and small hidden dimension?
  (2 points)

Results

Figure 1. 2-D latent space

Figure 2. 10-D latent space

Figure 3. 25-D latent space

Figure 4. 50-D latent space

Discussion

With vanilla VAE, we see that when the dimension of latent space is small e.g. H=2, VAE generates realistic images. However, it suffers from lack of diversity.

When H is increased to 10, the generation exhibits some greater variability but also begins to degrade in quality. As H is further increased to 25 and 50, the degradation continues.

Another difference we notice is that higher dimension latent space exhibits sharper images compared to low dimension.

Problem 3

- Perform a qualitative comparison the results of VAE according to the weight of KLD term in loss function for H = 2. (λ = 1; 5; 10; 40) In this problem, you NEED to sample the latent variables in a row for each axis as described in Figure 3. (2 points)

- What is the difference between VAEs with different KLD term weights? What is the advantage of increasing the weight of the KLD term in a situation where the data is more complex and the hidden dimension is larger? (3 points)

Results

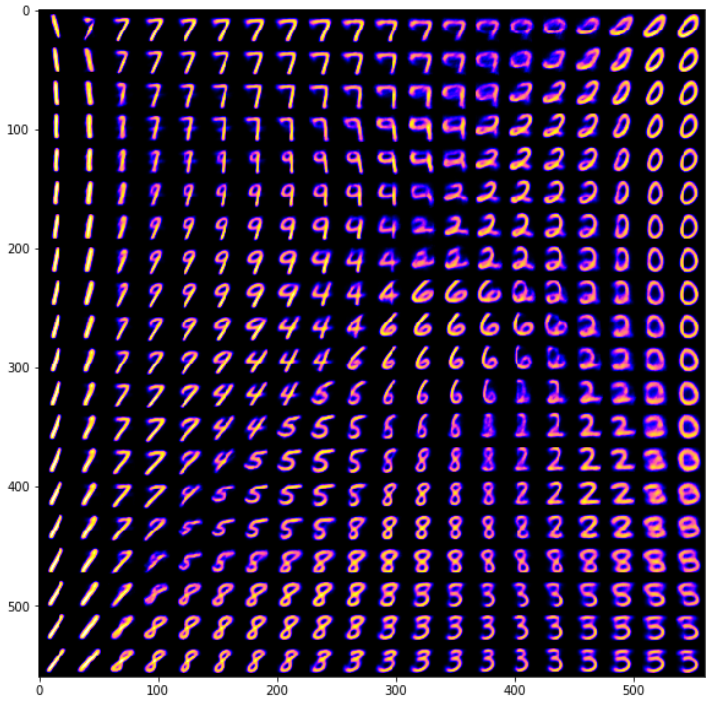

Figure 5. Output with weight = 1

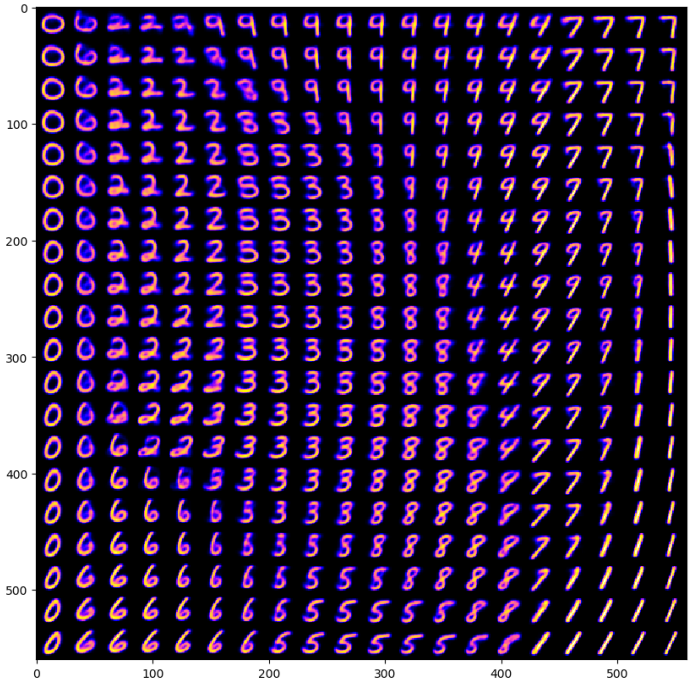

Figure 6. Output with weight = 5

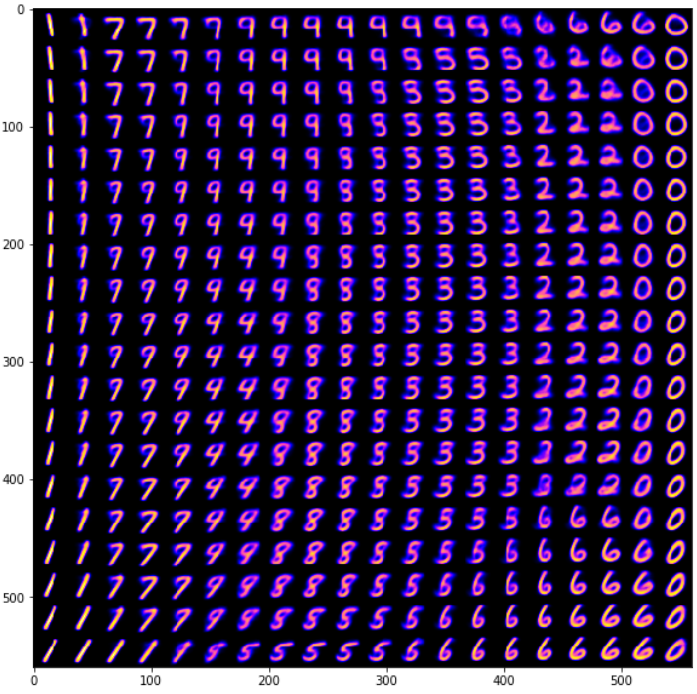

Figure 7. Output with weight = 10

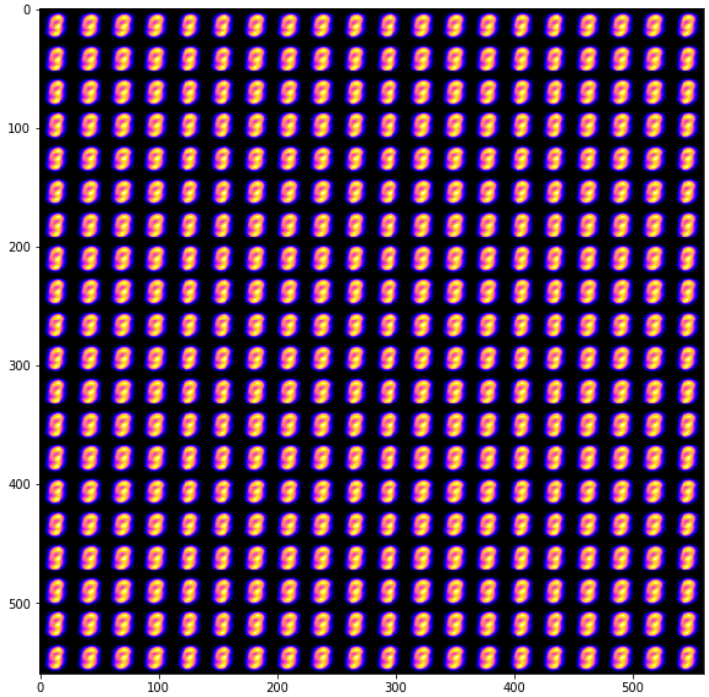

Figure 8. Output with weight = 40

Discussion

The intuition behind the weighted KLD term is that we want to make sure that the different neurons in our latent distribution are uncorrelated i.e. they all try to learn something different about the input data. This class of VAEs is known as Disentangled VAEs. This is implemented by adding a single hyper parameter λ to our loss function that weighs how much this KLD is present in the loss function. So in the disentangled version, the auto encoder will only use a specific latent variable if it really has a benefit and if it does not benefit the compression, then it will simply stick to the normal.

Varying λ changes the degree of applied learning pressure during training, thus encouraging different learnt representations. When λ = 1, it corresponds to the original VAE formulation. In order to learn disentangled representations it is important to set λ > 1, thus putting a stronger constraint on the latent bottleneck than in the original VAE formulation. The extra pressures coming from high β values, however, may create a trade-off between reconstruction fidelity and the quality of disentanglement within the learnt latent representations. By increasing λ, we are actually forcing our auto encoder to map the information onto only a few of latent variables.

Disentangled representations emerge when the right balance is found between information preservation (reconstruction cost as regularization) and latent channel capacity restriction (λ > 1). The latter can lead to poorer reconstructions due to the loss of high frequency details when passing through a constrained latent bottleneck.

When λ is too low or too high the model learns an entangled latent representation due to either too much or too little capacity in the latent **z** bottleneck. In general $\beta > 1$ is necessary to achieve good disentanglement. However if λ is too high and the resulting capacity of the latent channel is lower than the number of data generative factors, then the learnt representation necessarily has to be entangled (as a low-rank projection of the true data generative factors will compress

them in a non-factorial way to still capture the full data distribution well). We also note that VAE reconstruction quality is a poor indicator of learnt disentanglement. Good disentangled representations often lead to blurry reconstructions due to the restricted capacity of the latent information channel **z**, while entangled representations often result in the sharpest reconstructions. Therefore, one should not necessarily strive for perfect reconstructions when using λ -VAEs as unsupervised feature learners - though it is often possible to find the right λ -VAE architecture and the right value of λ to have both well-disentangled latent representations and good reconstructions

When the data is complex and the hidden dimension is larger, we can use larger λ to only use the latent variables that actually matter ignore the rest. This allows us to obtain an efficient representation of the data. The VAE will then consistently and robustly discovers more factors of variation in the data, and learn a representation that covers a wider range of factor values and is disentangled more cleanly. However, this approach relies on the optimization of hyper parameter λ, which can be found directly through a hyper parameter search if weakly labelled data is available.