Full name : Saad Benslimane
ID : 20228273
Email adress : saad.benslimane@umontreal.ca
April 14th, 2022

Problem 1

6 Using binarized MNIST dataset and Adam optimizer with learning rate of 3.10^{-4}, I trained the model on training set and evaluated it on validation set using ELBO for 20 epochs. From figure 1 we can see that : with KL divergence $D_{\mathrm{KL}}(q(z|x)\|p(z))$ via **analytic solution**, the **ELBO** decrease after each training to reach **101.33** at the end of the 20th epochs. While using KL diveregence $D_{\mathrm{KL}}(q(z|x)\|p(z))$ via **Monte Carlo**, this time the model takes more time for training (like 10 times more than using analytic solution) and the **ELBO** decrease after each training to reach **47.62** at the end of the 20th epochs.

7 For this question we found the **log-likelihood** of the trained VAE models on the test set is equal to **-95.64** using KL divergence $D_{\mathrm{KL}}(q(z|x)\|p(z))$ via **analytic solution**, and equal to **-292.07** using KL diveregence $D_{\mathrm{KL}}(q(z|x)\|p(z))$ via **Monte Carlo**.

Figure 1: VAE ELBO using KL divergence via analytic and Monte Carlo solutions with epochs=20, learning rate=3.10^{-4} and Adam optimizer.

Problem 2

2 **Blurriness** : From figure 2, we can see that the clarity of the generated images are not comparable to the authentic ones : Some of the generated pictures doesn't have the fine textures expected, and some of the generated images are not readable.
Diversity : Now on the variety side, we can see that the generated images reflect a good differences in color, style and variety of numbers.

Figure 2: Generated images from the trained generator.

3 For this question, I sampled a seed z as was asked. For the 100 latent variable dimensions I sampled every 15th of them and added 5 to the dimension. From 3 we can visualize the generated images : the first image represent the generated image using baseline z, and the next ones represent the generated images using $z's$ added by 5.

We observe clearly that the perturbations we added to the sample z have changed the shape of the image : they become more readable. Thus we conclude from this that the model had learned disentangled representations.

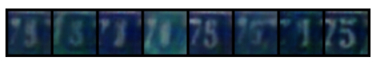

Figure 3: Generated images from the trained generator by adding perturbations to sample.

4.a For $\alpha = 0,\, 0.1,\, 0.2 \ldots 1$ and using : $z'_\alpha = \alpha z_0 + (1-\alpha)z_1$, figure 4 represent the resulting samples $x'_\alpha = g(z'_\alpha)$.

Figure 4: Generated samples by interpolating in the latent space.

4.b For $\alpha = 0,\, 0.1,\, 0.2 \ldots 1$ and using the data samples $x_0 = g(z_0)$ and $x_1 = g(z_1)$, figure 5 represent the resulting samples $\hat{x}_\alpha = \alpha x_0 + (1-\alpha)x_1$.

Figure 5: Generates samples by interpolating in the data space.

As we can see from the two figures, interpolating in the latent space gradually changed the number from an unclear 2 to a readable 8 in the final image. While interpolating in the data space doesn't change the number but gradually make it a more clear and a readable 8.

Problem 3

4 At first attempt I trained the Simsiam model using MLP='$fixed\_random\_init$' for both cases with and without stop gradient, as we can see from figures 6 and 7 : with stop gradient, the model converge to a loss value of -0.36 after 50 epochs. While without stop gradient, the mode converge to a loss value of -0.98 after 70 epochs. For Knn accuracy we can see clearly that for both cases we didn't get a good results and the accuracy fluctuates between 16% and 26%.
The second attempt I trained the Simsiam model using MLP='$None$' for both cases with and without stop gradient, and as we can see from figures 8 and 9 : this time we get a result similar to what we found in the article, for both cases the accuracy reach a value of 58% and 59% and the loss converge to a value of -0.64.

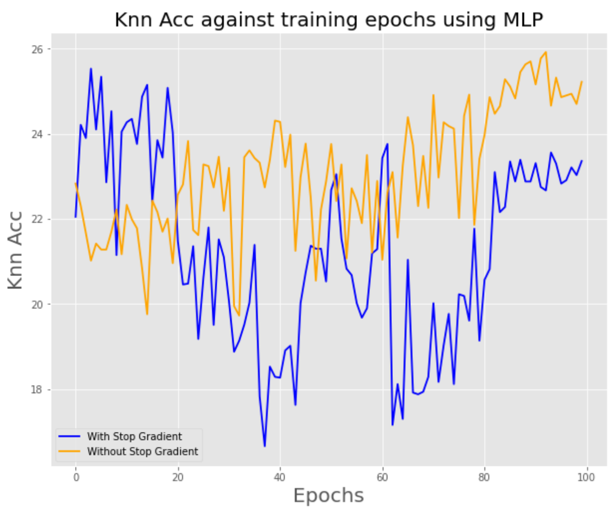

Figure 6: KNN Accuracy of a Simsiam trained for 100 epochs with and without stop gradient and MLP='$fixed\_random\_init$'.

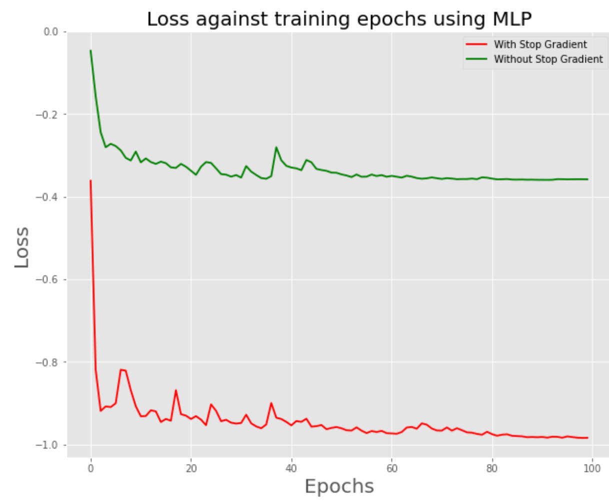

Figure 7: Loss of a Simsiam trained for 100 epochs with and without stop gradient and MLP='$fixed\_random\_init$'.

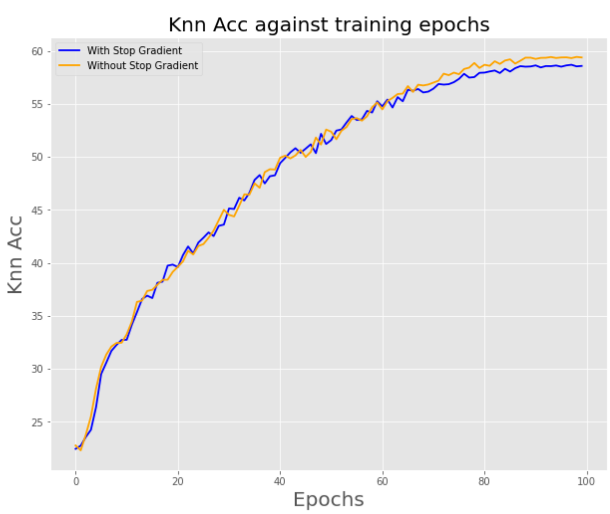

Figure 8: KNN Accuracy of a Simsiam trained for 100 epochs with and without stop gradient and MLP='$None$'.

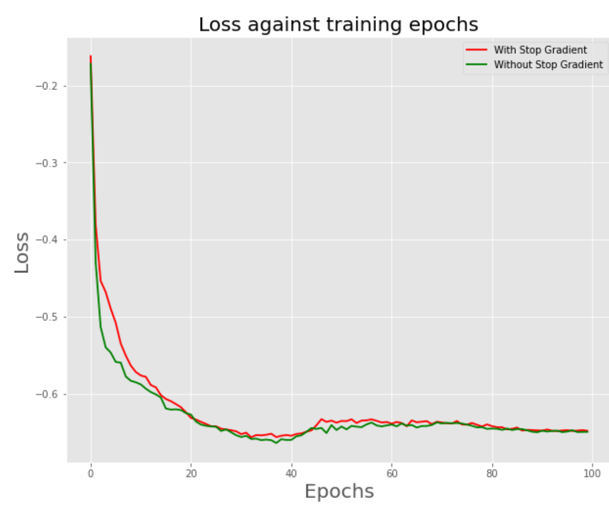

Figure 9: Loss of a Simsiam trained for 100 epochs with and without stop gradient and MLP='$None$'.

5 For this question I replaced the MLP by a Identity mapping. From figures 10 and 11 we can see that, for both

cases, the accuracy keep decreasing to a value of 10% (with stop gradient the model doesn't fluctuate a lot). While the loss reach is almost equal to -1 for both cases.

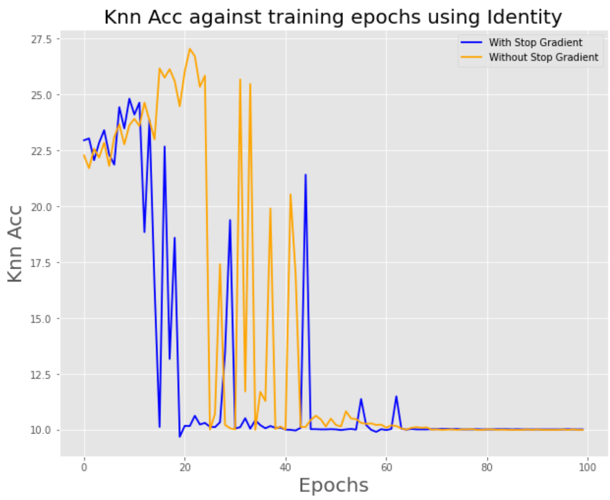

Figure 10: KNN Accuracy of a Simsiam trained for 100 epochs with and without stop gradient and MLP='*no_pred_mlp*'.

Figure 11: Loss of a Simsiam trained for 100 epochs with and without stop gradient and MLP='*no_pred_mlp*'.