CSI5138 Assignment4

Name: Jiacheng Hou
Student Number: 300125708

In the assignment, I aim to implement and train three models: VAE, GAN and WGAN, for the MNIST and CIFAR10 datasets. I also explore how model complexity and latent vector dimension affect generated images. Firstly, I try two latent vector dimensions in VAE model for MNIST dataset. Secondly, I try two models of VAE for CIFAR10 dataset, which are linear layers for encoding and decoding and convolutional and transposed-convolutional layers for encoding and decoding. Thirdly, I try two models of WGAN for MNIST dataset, one is linear layers for generator and critic, and another one is convolutional and transposed-convolutional layers for generator and critic.

1 MNIST dataset

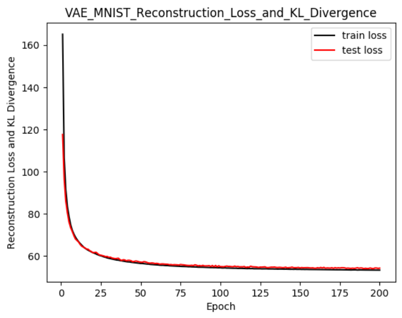
In this section, all models trained on the MNIST dataset are explained. Table 1 and Table 2 show generated images for each model.

1.1 VAE

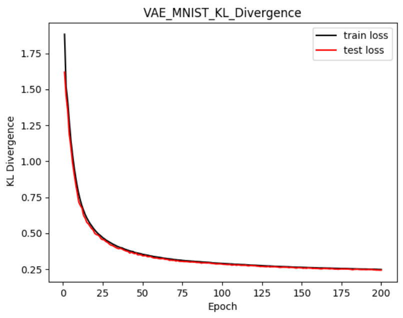
In this section, I try two latent vector dimensions, 128 dimensions and 2 dimensions, in VAE model for MNIST dataset. I find that with the increase of latent vector dimensions, the model performs better.

1.1.1 VAE – latent vector dimension: 128

In the VAE model, I have 6 linear layers in total. The first three layers are encoding layers and the last three layers are decoding layers. The latent vector dimension is 128. The loss function is the Binary Cross Entropy loss. It will be used to calculate the reconstruction loss. The total loss is calculated by adding the reconstruction loss and the KL divergence. I train and test the model for 200 epochs. Figure 1a shows the training dataset and testing dataset total losses change over epochs. Figure 1b shows the training dataset and testing dataset KL divergence change over epochs.



(a) Training dataset and testing dataset total losses against epochs

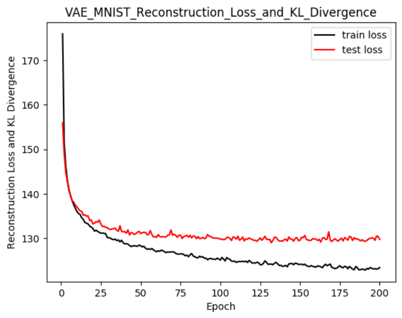


(b) Training dataset and testing dataset KL divergence against epochs

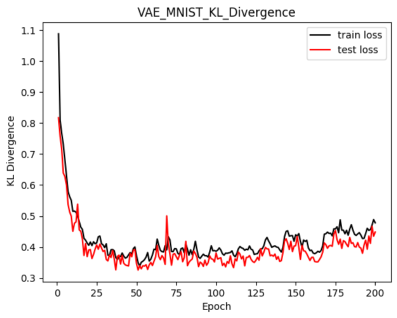
Figure 1: VAE model (Latent vector dimension: 128) on MNIST

1.1.2 VAE – latent vector dimension: 2

This section uses the same model and hyper-parameters as the above section except for the latent vector dimension, which is 2. Figure 2a shows the training dataset and testing dataset total losses change over epochs. Figure 2b shows the training dataset and testing dataset KL divergence change over epochs.



(a) Training dataset and testing dataset total losses against epochs

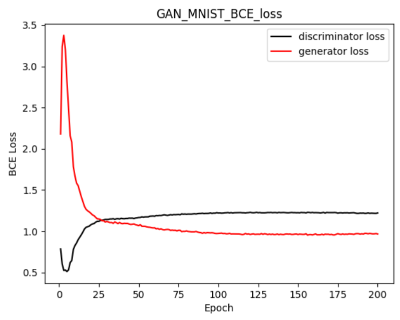


(b) Training dataset and testing dataset KL divergence against epochs

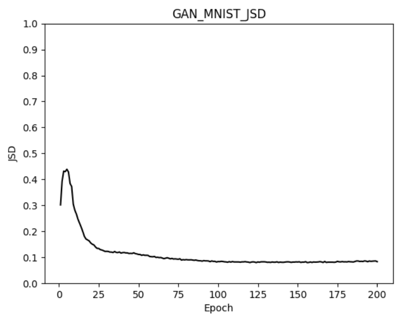
Figure 2: VAE model (Latent vector dimension: 2) on MNIST

1.2 GAN

In the GAN model, I have 4 linear layers and LeakyReLU activations in between as the generator neural network. For the discriminator neural network, I have 4 linear layers, LeakyReLU activations and Dropout in between, and the Sigmoid activation layer at last. The latent vector dimension is 128. The loss function is the Binary Cross Entropy loss. I train the model for 200 epochs. Figure 3a shows the generator and discriminator losses change over epochs. Figure 3b shows the JS Divergence changes over epochs.



(a) Generator and discriminator losses



(b) Jensen–Shannon Divergence

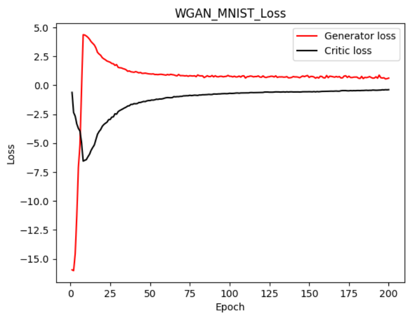
Figure 3: GAN model on MNIST

1.3 WGAN

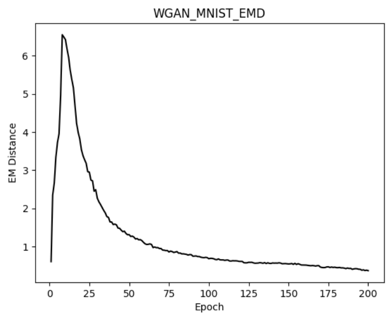
In this section, I try two models for WGAN, one is WGAN, and the other one is DC-WGAN. The DC-WGAN model performs better than WGAN on MNIST dataset.

1.3.1 WGAN

In the WGAN model, I have 4 linear layers and LeakyReLU activations in between as the generator neural network. For the critic neural network, I have 4 linear layers and LeakyReLU activations and Dropout in between. The latent vector dimension is 128. The weight clipping is 0.01. I train the model for 200 epochs. Also, I train the critic 5 times and then train the generator. Figure 4a shows the generator and critic losses change over epochs. Figure 4b shows the EM distance changes over epochs.



(a) Generator and critic losses

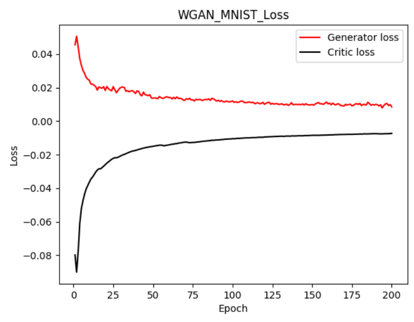


(b) Earth Mover's distance

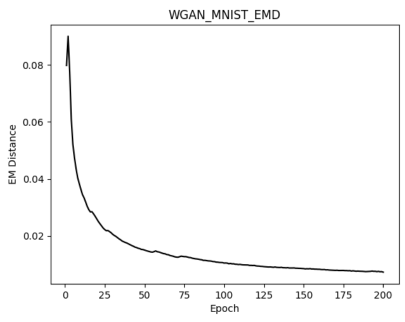
Figure 4: WGAN model on MNIST

1.3.2 DC-WGAN

In the DC-WGAN model, I have convolutional layers for the discriminator and transposed-convolutional layers for the generator. Both the discriminator and generator include batch norm layers. While the generator uses ReLu activation layers and the discriminator uses LeakyReLu activation layers. The discriminator's last layer is the Sigmoid activation layer. The latent vector dimension is 128. The weight clipping is 0.01. I train the model for 200 epochs. Also, I train the critic 5 times and then train the generator. Figure 5a shows the generator and discriminator losses change over epochs. Figure 5b shows the JS Divergence changes over epochs.



(a) Generator and critic losses



(b) Earth Mover's distance

Figure 5: DC-WGAN model on MNIST

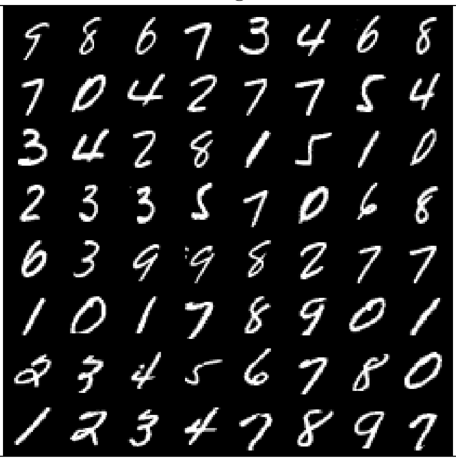
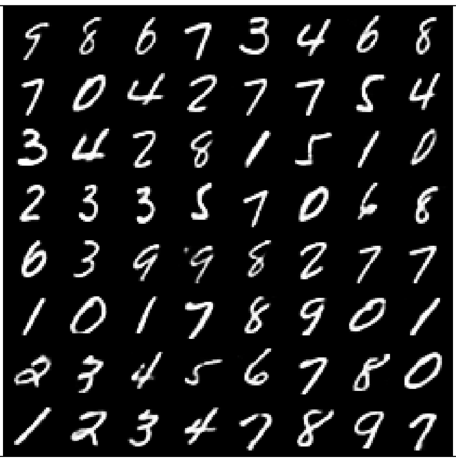
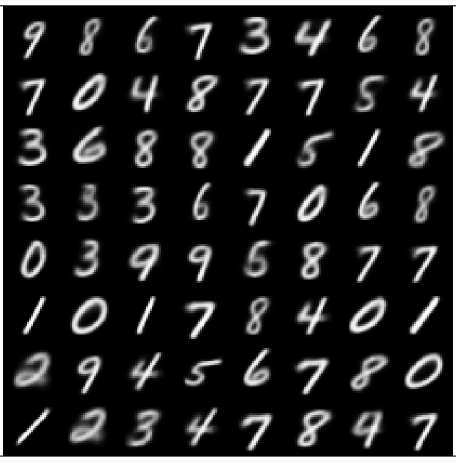
Model	Images
Samples	
• VAE – Epoch: 200 – Latent vector dimension: 128 – Encoding: 3 Linear layers – Decoding: 3 Linear layers	
• VAE – Epoch: 200 – Latent vector dimension: 2 – Encoding: 3 Linear layers – Decoding: 3 Linear layers	

Table 1: MNIST images results I

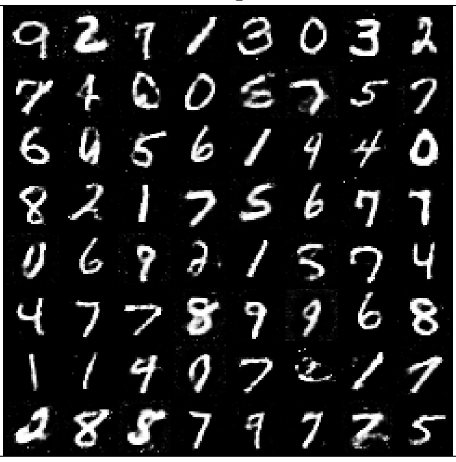
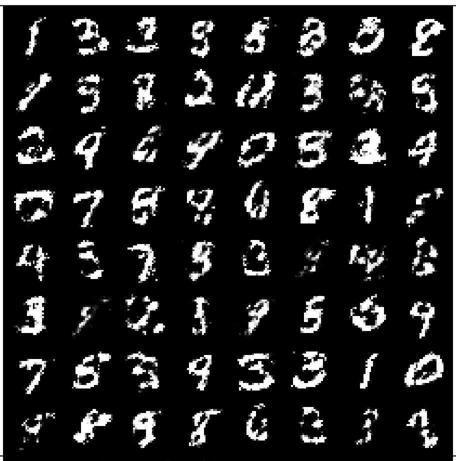
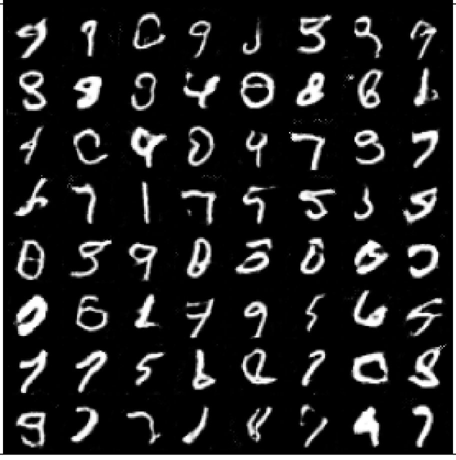
Model	Images
• GAN – Epoch: 200 – Latent vector dimension: 128 – Generator: 4 Linear layers and LeakyReLU activations in-between – Discriminator: 4 Linear layers, LeakyReLU activations and Dropout in-between, Sigmoid activation layer at last	
• WGAN – Epoch: 200 – Latent vector dimension: 128 – Generator: 4 Linear layers and LeakyReLU activations in-between – Critic: 4 Linear layers, LeakyReLU activations and Dropout in-between	
• DC-WGAN – Epoch: 200 – Latent vector dimension: 128 – Generator: Transposed convolutional layers, batch norm and ReLU in between – Critic: Convolutional layers, batch norm and LeakyReLU in between	

Table 2: MNIST images results II

2 CIFAR10 dataset

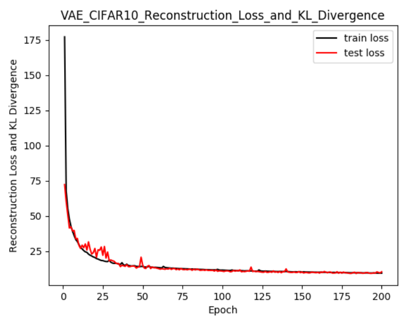
In this section, all models trained on CIFAR10 dataset are explained. Table 3 and Table 4 show generated images for each model.

2.1 VAE

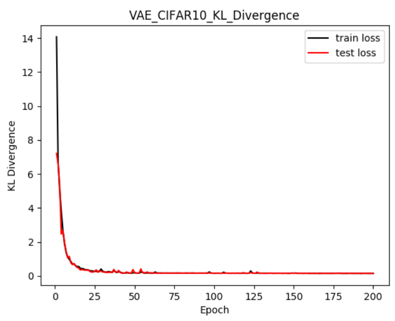
I try two models in VAE, one is linear layers for encoding and decoding and another one is convolutional and transposed-convolutional layers. Compared with linear layers for encoding and decoding, convolutional and transposed-convolutional layers for encoding and decoding performs much better for the CIFAR10 dataset.

2.1.1 VAE – encoding and decoding: convolutional and transposed-convolutional layers

In the VAE model, I have convolutional layers in the encoding part and transposed-convolutional layers in the decoding part. Both the encoding and decoding layers include batch norm layers. The latent vector dimension is 512. The loss function is the Mean Squared Error. It will be used to calculate the reconstruction loss. The total loss is calculated by adding the reconstruction loss and the KL divergence. I train and test the model for 200 epochs. Figure 6a shows the training dataset and testing dataset total losses change over epochs. Figure 6b shows the training dataset and testing dataset KL divergence change over epochs.



(a) Training dataset and testing dataset total losses against epochs

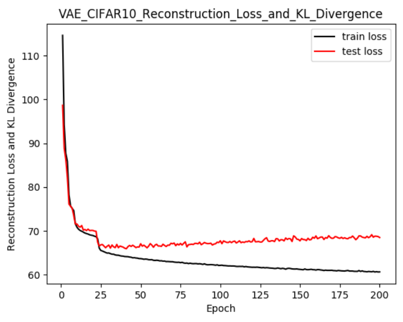


(b) Training dataset and testing dataset KL divergence against epochs

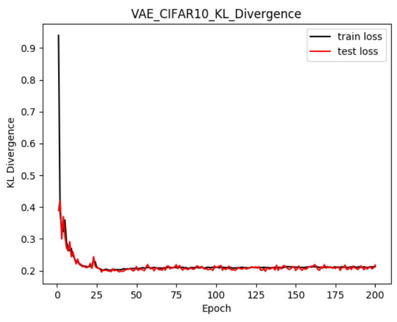
Figure 6: VAE model (convolutional and transposed-convolutional layers) on CIFAR10

2.1.2 VAE – encoding and decoding: linear layers

In the VAE model, I have 6 linear layers in total. The first three layers are encoding layers and the last three layers are decoding layers. The latent vector dimension is 512. The loss function is the Mean Squared Error. It will be used to calculate the reconstruction loss. The total loss is calculated by adding the reconstruction loss and the KL divergence. I train and test the model for 200 epochs. Figure 7a shows the training dataset and testing dataset total losses change over epochs. Figure 7b shows the training dataset and testing dataset KL divergence change over epochs.



(a) Training dataset and testing dataset total losses against epochs

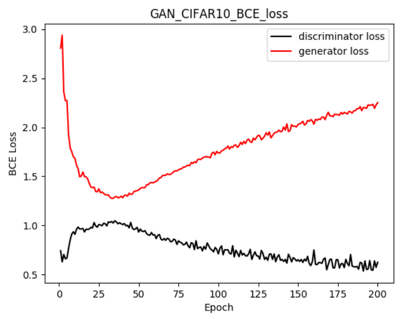


(b) Training dataset and testing dataset KL divergence against epochs

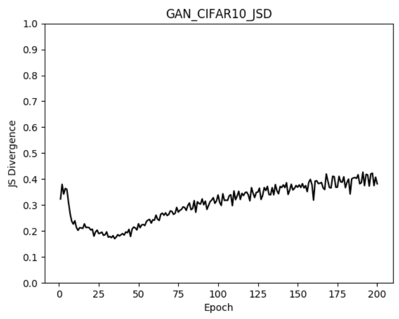
Figure 7: VAE model (linear layers) on CIFAR10

2.2 DCGAN

In the DCGAN model, I have convolutional layers for the discriminator and transposed-convolutional layers for the generator. Both the discriminator and generator include batch norm layers. While the generator uses ReLu activation layers and the discriminator uses LeakyReLu activation layers. The discriminator's last layer is the Sigmoid activation layer. The latent vector dimension is 128. The loss function is the Binary Cross Entropy loss. I train the model for 200 epochs. Figure 8a shows the generator and discriminator losses change over epochs. Figure 8b shows the JS Divergence changes over epochs.



(a) Generator and discriminator losses

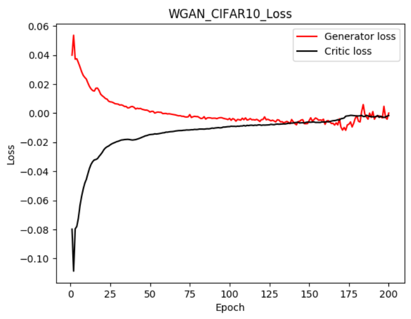


(b) Jensen–Shannon Divergence

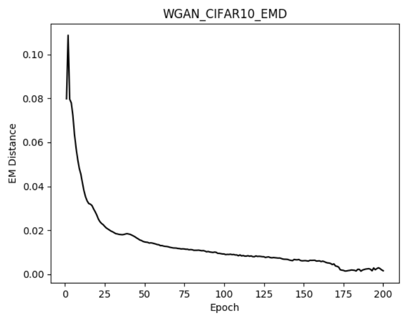
Figure 8: DCGAN model on CIFAR10

2.3 DC-WGAN

In the DC-WGAN model, I have convolutional layers for the discriminator and transposed-convolutional layers for the generator. Both the discriminator and generator include batch norm layers. While the generator uses ReLu activation layers and the discriminator uses LeakyReLu activation layers. The latent vector dimension is 128. The weight clipping is 0.01. I train the model for 200 epochs. Also, I train the critic 5 times and then train the generator. Figure 9a shows the generator and critic losses change over epochs. Figure 9b shows the EM distance changes over epochs.



(a) Generator and critic losses



(b) Earth Mover's distance

Figure 9: DC-WGAN model on CIFAR10

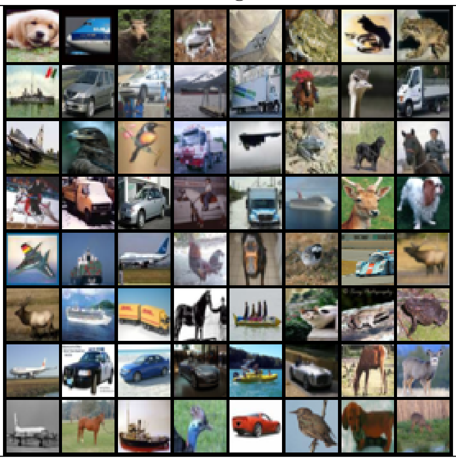
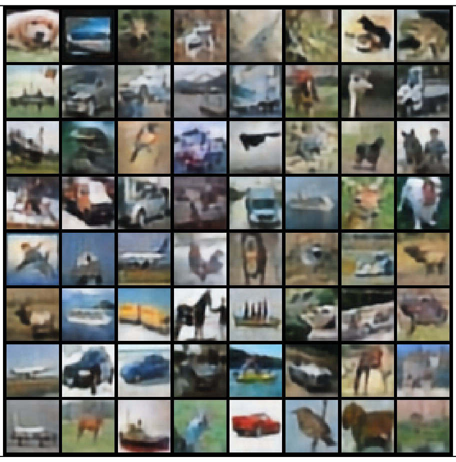
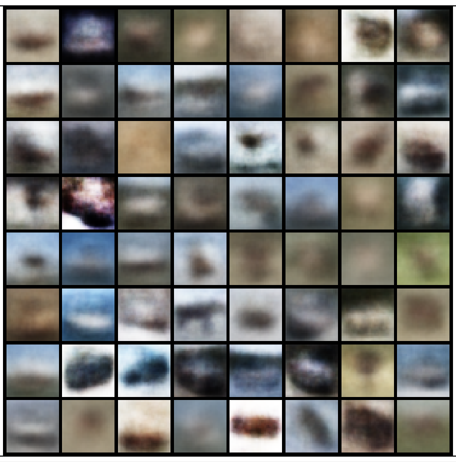
Model	Images
Samples	
• VAE – Epoch: 200 – Latent vector dimension: 512 – Encoding: Convolutional layers and batch norm layers in between – Decoding: Transposed convolutional layers and batch norm layers in between	
• VAE – Epoch: 200 – Latent vector dimension: 512 – Encoding: 3 Linear layers – Decoding: 3 Linear layers	

Table 3: CIFAR10 images results I

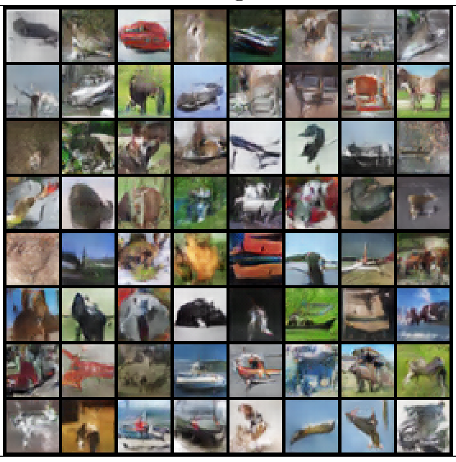
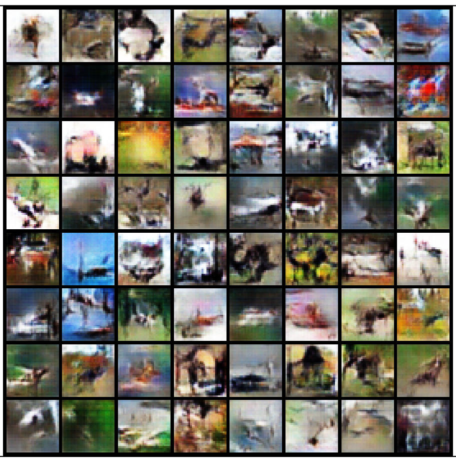
Model	Images
• DCGAN – Epoch: 200 – Latent vector dimension: 128 – Generator: Transposed convolutional layers, batch norm and ReLU in between – Discriminator: Convolutional layers, batch norm and LeakyReLU in between, Sigmoid at last	
• DC-WGAN – Epoch: 200 – Latent vector dimension: 128 – Generator: Transposed convolutional layers, batch norm and ReLU in between – Critic: Convolutional layers, batch norm and LeakyReLU in between	

Table 4: CIFAR10 images results II