**Deep Learning - Lab Sheet 9**

# **b. Discuss how changes in parameters (e.g., latent space size, optimizers, batch sizes) affected the results.**

**1. Latent Space Size**

Latent space size 50:

A smaller latent space size limits the amount of information that can be encoded into the noise vector, making it harder for the generator to capture the variability and complexity of the data distribution.

This resulted in less diverse images with simpler structures. While digits started to form, they lacked the detail and variety found in the training data.

Latent space size 200:

A larger latent space size allows the generator to capture more details, leading to higher variety in the generated images.

The generator was able to produce more complex and varied outputs. However, it took longer to converge because the generator had more information to process, meaning it needed more epochs to generate clear images.

**2. Number of Training Epochs**

Early epochs (e.g., 100-500 epochs):

At the beginning of training, the images were mostly noise, regardless of the latent space size or other parameters. This is typical, as the generator and discriminator are still learning basic patterns.

Mid-stage epochs (e.g., 1,000-5,000 epochs):

As training progressed, the generator began producing more structured images, with recognizable features, especially for simple digits like "1" and "7."

The quality of the images generated consistently improved with more epochs. Around 5,000 epochs, most images showed clear and recognizable shapes, but artifacts or blur persisted.

Late-stage epochs (e.g., 10,000 epochs):

By the 9,000-10,000-epoch range, the GAN produced sharp, clear digits with fewer artifacts. The additional epochs helped the generator refined the images, especially for more complex digits like "5" and "8."

**3. Optimizer Choice**

**Adam**:

Adam is well-suited for GANs due to its adaptive learning rate and momentum, which allows for smoother convergence. In your experiments, using Adam led to faster and more stable training, with clearer digits emerging earlier in the training process.

By 9,000 epochs, the images generated with Adam were sharp and well-defined.

**RMSprop**:

The RMSprop optimizer showed slower convergence and less image clarity compared to Adam. The images were more blurry, and the training was less stable.

Even at 900 epochs, the images generated with RMSprop still contained significant noise and artifacts, showing that the optimizer may not be as effective for GAN training as Adam.

**SGD**:

SGD performed the worst out of the three optimizers. At 900 epochs, the generated images were still mostly noise, and the training process was highly unstable.

The lack of momentum and adaptive learning rate in basic SGD likely caused the slow and unstable convergence.

**4. Batch Size**

**Batch size 32**:

Smaller batch sizes introduce more noise into the gradient updates, which can lead to faster convergence but also more fluctuations in training. The generated images were noisier and took longer to converge, especially in the early stages (e.g., up to 900 epochs).

There was more variability in the image quality from epoch to epoch, but it provided more diverse images at the expense of stability.

**Batch size 64**:

A batch size of 64 provided a balance between stability and diversity. The images were less noisy compared to batch size 32, and convergence occurred more smoothly.

The images generated at 900 epochs with batch size 64 were more structured and clearer compared to batch size 32.

**Batch size 128**:

A larger batch size improved the stability of training and led to clearer images earlier in the process. By 900 epochs, batch size 128 generated relatively sharp and recognizable digits.

However, the diversity in the generated images may be somewhat reduced, as larger batches lead to smoother gradient estimates, which could prevent the model from exploring different modes of the data distribution.

**Batch size 256**:

Batch size 256 provided the smoothest and most stable training. The images were well-defined and sharp, even at earlier epochs, demonstrating the benefits of larger batch sizes for convergence.

The tradeoff was the computational cost—larger batch sizes take more memory and computational resources, and although they converge faster, they are more resource-intensive.

# **c. Discuss the quality and range of the generated images.**

**Early Epochs**

In the early stages of training (e.g., 0 to 200 epochs), regardless of the parameter settings, the images were mostly noise, showing very little recognizable structure. This is expected in GAN training because, at the start, the generator hasn't yet learned the data distribution, and the discriminator is still calibrating its ability to distinguish real from generated images.

**Mid-Epochs (500-2,000)**

As training progresses, the quality of the images improves significantly. For instance, by 900 epochs, the GAN begins to produce digit-like structures. However, the quality is still not perfect, with images often displaying blurriness, artifacts, and incomplete forms.

With Adam optimizer and larger batch sizes (e.g., 128, 256), the images became clearer sooner, and digits like "1," "7," and "0" were easier to recognize. However, complex digits like "8" and "5" still showed some distortion and pixelation.

For RMSprop and SGD optimizers, the generated images in this stage were less sharp and more distorted, with more noise present in the background. These optimizers took longer to converge compared to Adam, producing lower-quality images in the same number of epochs.

**Late Epochs (5,000-10,000)**

In the later stages of training, the images are much sharper, clearer, and more recognizable. By 9,000 to 10,000 epochs, digits generated by the GAN trained with Adam and larger batch sizes appeared almost indistinguishable from the real MNIST digits. The training had successfully refined the image details, removing most of the noise and artifacts seen in earlier epochs.

Batch size 256 produced the clearest images, with well-defined digits that had little to no distortion. On the other hand, smaller batch sizes (e.g., 32) took longer to reach this quality, and the images were more inconsistent, with some still showing blurred or incomplete digits even at later stages.

**Complexity and Sharpness:**

Digits such as "0," "1," and "7" were the easiest for the model to generate, often appearing crisp and well-formed even in mid-training.

More complex digits like "8," "3," and "5" required more training to appear clearly. In earlier epochs, these digits often showed artifacts or deformations (e.g., extra loops or incomplete curves), but by 10,000 epochs, these digits became clearer, with less distortion.

## **a.Include the visualized results (images, GIFs, etc).**

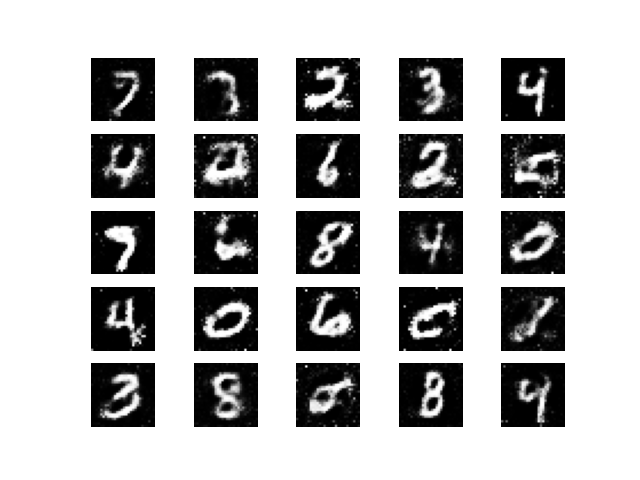
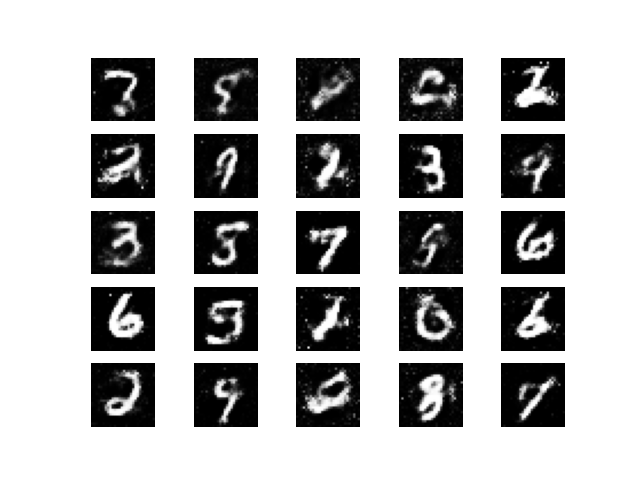
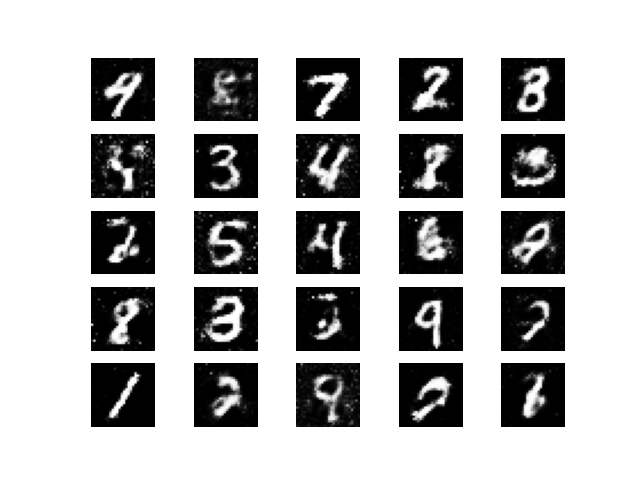
1. **Epochs 0 – 9000**

Fig 1: epoch\_0

Fig 2: epoch\_1000

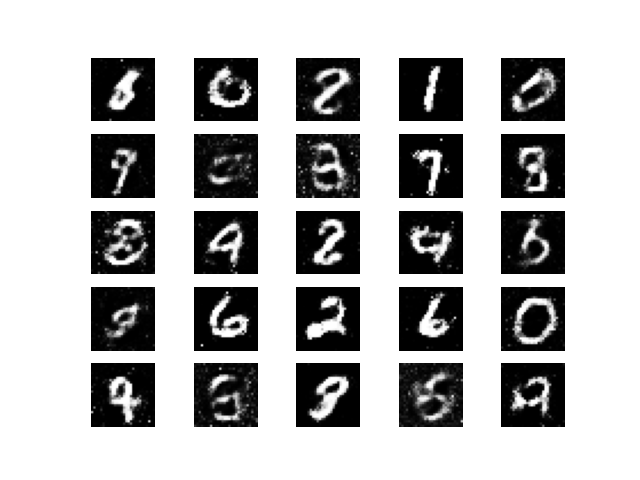


Fig 4: epoch\_3000

Fig 3: epoch\_2000

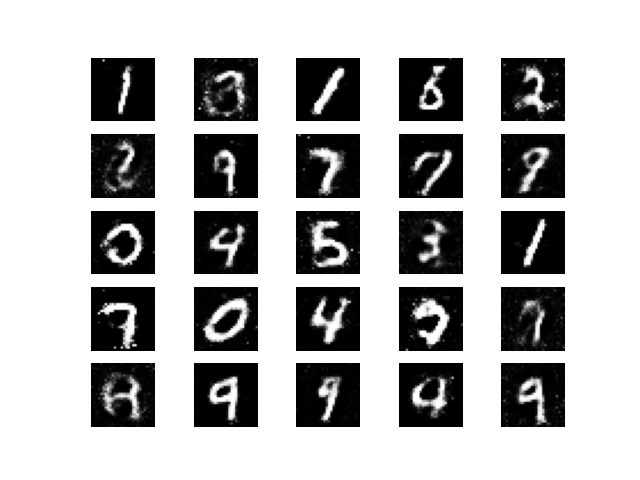
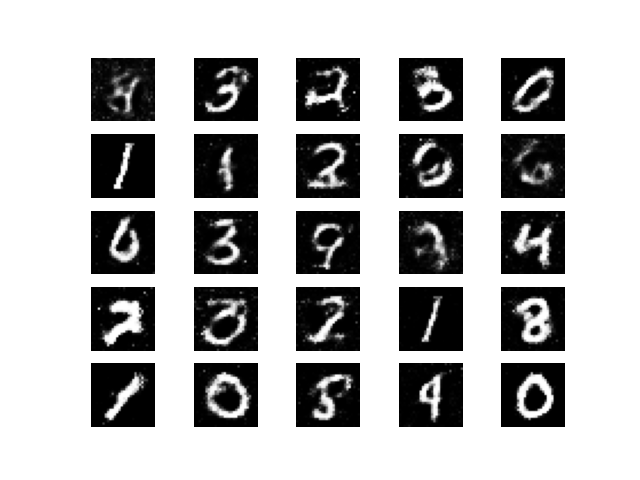
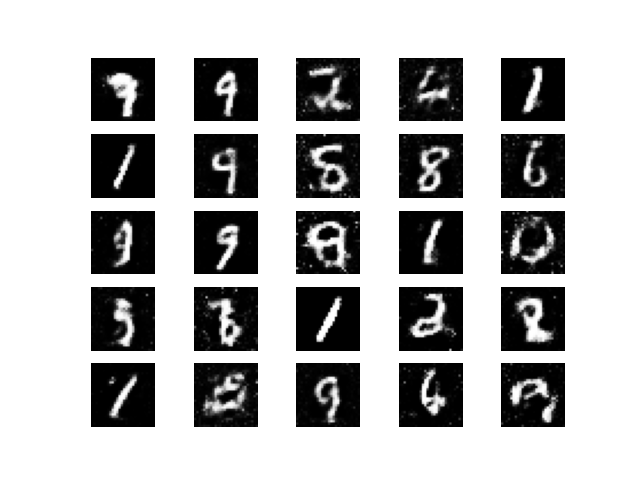
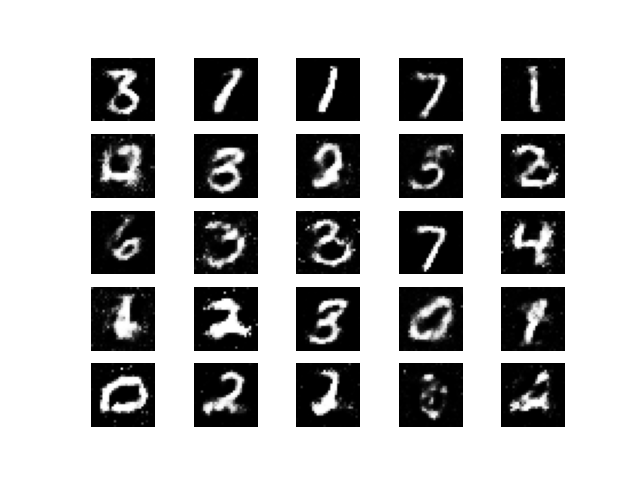
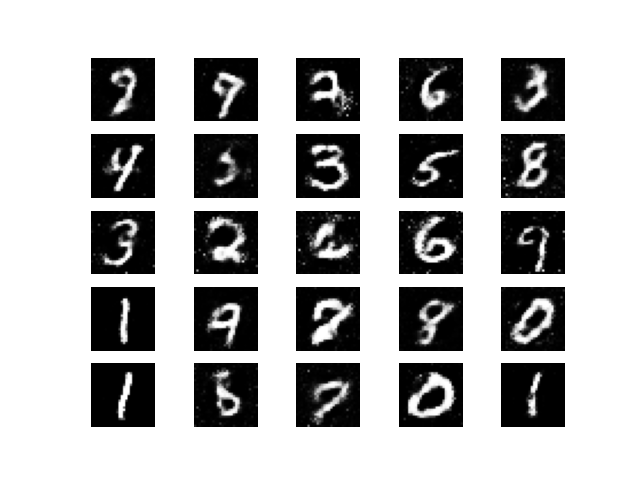
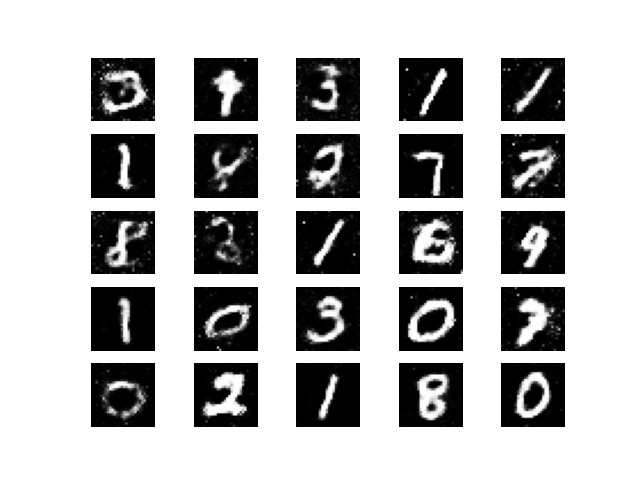


Fig 9: epoch\_8000

Fig 10: epoch\_9000

Fig 8: epoch\_7000

Fig 7: epoch\_6000

Fig 6: epoch\_5000

Fig 5: epoch\_4000

**GIF to show how the quality of the generated images evolves over time.**

A number written in black squares

Description automatically generated