**DEEP LEARNING PROJECT ASSIGNTMENT REPORT 2**

-Sri Sakticharan Nirmal kumar (1337576)

**1.BASELINE MODEL: IMPACT OF NUMBER OF EPOCHS ON IMAGE QUALITY**

**Baseline Model Results:**

A number in a square

Description automatically generated with medium confidence A number in a square

Description automatically generated with medium confidence

1. Epoch – 10 B. Epoch – 50

A number in a square

Description automatically generated

C. Epoch – 100

**Observation**:

**Epoch 10 -**

• **Clarity**: The generated digits are faint and poorly defined. The shapes are indistinct, and significant noise is present.

• **Noise**: There are irregular shapes and artifacts in the outputs that do not resemble digits well.

**Epoch 50 -**

• **Clarity**: The digits become more defined compared to epoch 10. The outlines of many digits are noticeably sharper.

• **Noise**: The noise present in earlier epochs is reduced significantly, and the generated outputs are more structured.

**Epoch 100 -**

• **Clarity**: The digits are sharp and well-structured, resembling realistic handwritten digits.

• **Noise**: The noise is minimal, with smooth and clean outputs.

**Key Takeaway:**

1. **Improvement Over Epochs**:

• A noticeable improvement in clarity, diversity, and reduction of noise is observed as training progresses.

• By epoch 100, the generated outputs closely resemble real handwritten digits.

2. **Stability**:

• The early stages of training (e.g., epoch 10) show significant instability, with poorly formed and inconsistent outputs.

• By epoch 100, the model stabilizes and produces high-quality, realistic digits.

**2. RELU ACTIVATION VS LEAKYRELU ACTIVATION**

A number in a square

Description automatically generated with medium confidence A number in a square

Description automatically generated with medium confidence

D. Epoch – 50 (LeakyReLU, baseline) E. Epoch – 50 (ReLU)

**Observations:**

**Baseline (LeakyReLU) - Epoch 50**

• **Clarity**: The digits are well-formed, and sharper compared to earlier epochs. However, a few digits still have faint or inconsistent strokes.

• **Noise**: Minimal noise is observed, though some artifacts (e.g., incomplete strokes) persist.

**Modified Model (ReLU) - Epoch 50**

• **Clarity**: The digits are noticeably sharper compared to the baseline. The strokes are more defined, and the digits appear more realistic.

• **Noise**: Reduced noise compared to the baseline, resulting in smoother digit shapes and clearer outputs.

**Key Takeaways:**

1. **ReLU vs. LeakyReLU**:

• ReLU activation results in sharper and more realistic digit generation at epoch 50.

• The clarity of outputs is slightly better with ReLU, likely due to its strong activation behavior compared to the more gradual slope of LeakyReLU.

2. **Noise Reduction**:

• ReLU leads to smoother outputs with fewer artifacts, indicating better convergence during training.

**BASELINE VS MODIFIED PARAMETERS**

Change hyperparameters:

* the dimensionality of the noise vector,
* the batch size,
* the learning rates, and
* the momentum terms.

**Old parameters VS New Parameters**

|  |  |  |
| --- | --- | --- |
| **PARAMETER** | **BASELINE** **VALUES** | **MODIFIED NEW PARAMETERS** |
| Noise Vector Dimension | 100 | 200 |
| Batch Size | 64 | 256 |
| Learning Rate | 1e-4 | 1e-3 |
| Momentum | 0.9 (default) | 0.7 |

**Justification for new parameters values:**

**Noise Vector Dimension (200) :** Increasing the dimensionality of the noise vector allows the generator to explore a more diverse latent space, leading to greater variability and richer detail in the generated images. This helps in producing more diverse and realistic outputs.

**Batch Size (256)**: A larger batch size improves the stability of gradient updates during training. This reduces variance in updates and enhances the model’s ability to converge, although it increases computational requirements.

**Learning Rate (1e-3):** A higher learning rate accelerates the convergence process. However, it requires careful tuning to avoid overshooting the optimal solution. This value strikes a balance between convergence speed and stability.

**Momentum (0.7) :**A lower momentum value allows for more responsive weight updates. While this introduces slightly more noise, so as to check this I tried a lesser learning rate.

**Output Results :**

A number in a square

Description automatically generated with medium confidence A group of white letters in black squares

Description automatically generated

F. Epoch – 50 (LeakyReLU, Baseline) G. Epoch – 50(New Parameters)

**Observations: Baseline vs. Modified Parameters**

**Baseline:**

• **Clarity**: The digits are reasonably well-formed and clearly identifiable. However, some digits have faint strokes or incomplete shapes.

• **Noise**: Noise levels are low, but occasional artifacts (e.g., smudged areas or overlapping strokes) are present.

**Modified Parameters:**

• **Clarity**: The digits are less clear compared to the baseline. Many digits appear faint or poorly structured, and some shapes are difficult to identify.

• **Noise**: Noise levels are higher, leading to distorted or incomplete digits (e.g., 1, 2, and 8).

**Key Takeaways:**

**Impact of Modified Parameters**:

• While the modified parameters aimed to accelerate convergence and improve diversity, they appear to have introduced instability into the training process.

• A higher learning rate (1e-3) and reduced momentum (0.7) may have caused overshooting during optimization, leading to less consistent outputs.

**Comparison to Baseline**:

• The baseline parameters produce clearer and more stable outputs at the same epoch, indicating better overall performance.

• The modified parameters result in noisier and less diverse outputs, suggesting the need for further tuning (e.g., reducing the learning rate or increasing momentum).

**CONCLUSION**

In this project, we implemented a baseline DCGAN model and evaluated the impact of increasing epochs on the quality of generated images. We modified the generator’s activation function to use ReLU, analyzing its effect on clarity and training stability. Additionally, we experimented with hyperparameter tuning by altering the noise vector dimension, batch size, learning rate, and momentum. These experiments allowed us to observe how each change influenced the diversity, clarity, and stability of the generated outputs, providing insights into optimizing GAN training.