CS 599: Deep Learning - Lab 4

Neha Nagaraja
School of Informatics, Computing, and Cyber Systems
Northern Arizona University
`nn454@nau.edu`

Link to the Github: https://github.com/Neha-Nagaraja/CS599-DL/tree/master/Assignment5

1 Problem 1: Implementing various Recurrent neural network cells using basic tensorflow ops

1.1 Objective

In this project, I implemented two recurrent neural network (RNN) cells — the Gated Recurrent Unit (GRU) and the Minimal Gated Unit (MGU) — using basic TensorFlow operations. My goal was to compare the performance of these two units in solving a temporal credit assignment task on the notMNIST small dataset.

1.2 Implementation

About the Dataset The notMNIST small dataset contains grayscale images of the letters A–J from different fonts. The images are of size 28x28 pixels. The dataset includes some noise, with a small percentage of mislabeled images. I manually loaded and preprocessed the images, normalizing pixel values to the range [0, 1].

I implemented both GRU and MGU cells from scratch using TensorFlow Variables.

The GRU unit uses update and reset gates to control information flow.

The MGU unit simplifies this further by using only a forget gate.

I manually implemented all hidden-to-hidden, input-to-hidden, and output computations without relying on Keras built-in layers.

Experimental Setup

- Number of Hidden Units: 128
- Batch Size: 128
- Number of Trials: 3
- Number of Epochs per Trial: 10
- Optimizer: Adam

For each model (GRU and MGU), I trained across 3 trials and reported the average classification error and accuracies.

Below are the training vs test loss curves for each model:

Here is the test accuracy GRU vs MGU

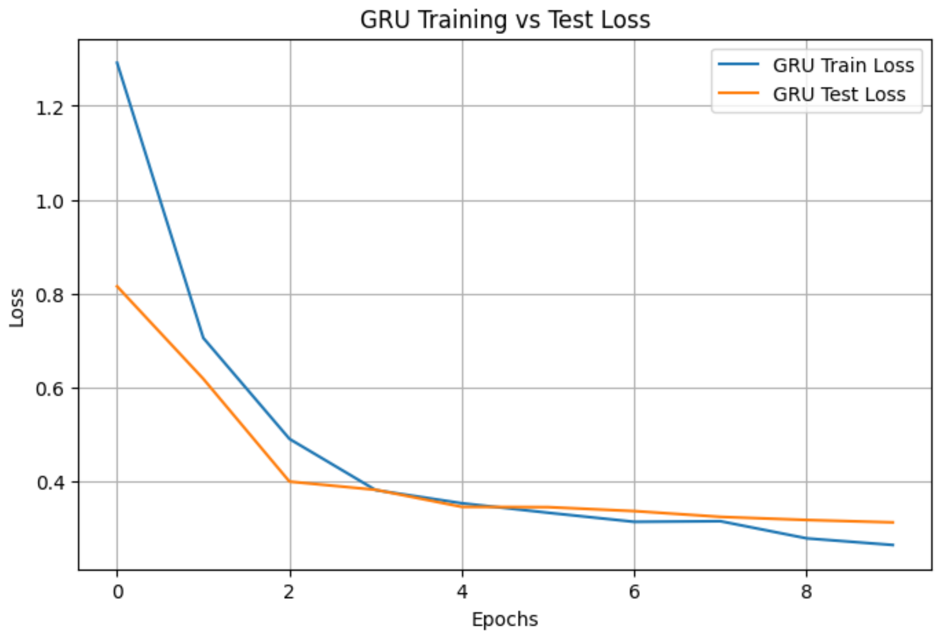

Figure 1: GRU Training vs test loss curves

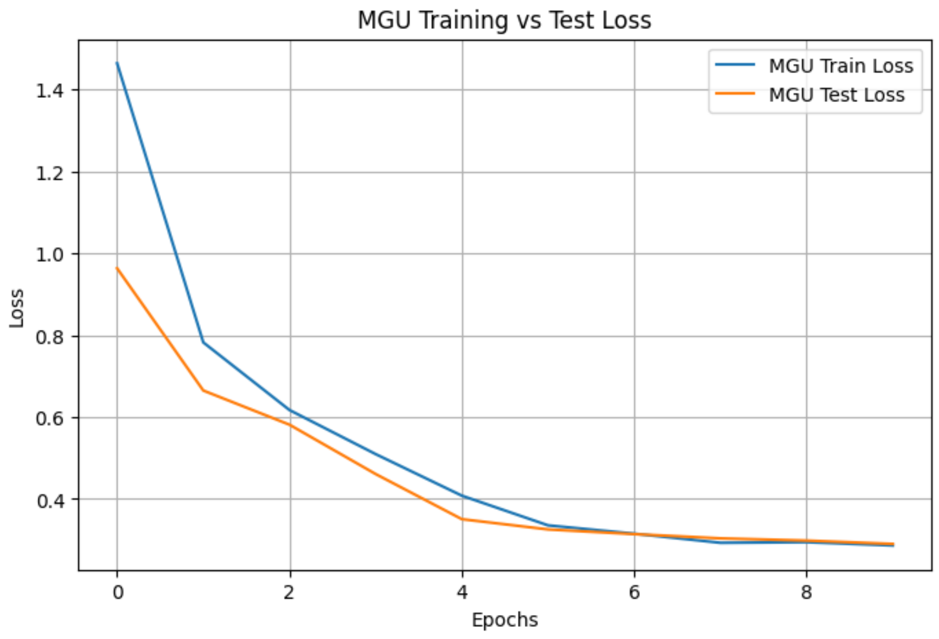

Figure 2: MGU Training vs test loss curve

Model	Average Training Accuracy	Average Test Accuracy	Average Classification Error
GRU	91.15%	89.86%	10.14%
MGU	91.64%	91.69%	8.31%

Table 1: Performance comparison between GRU and MGU models.

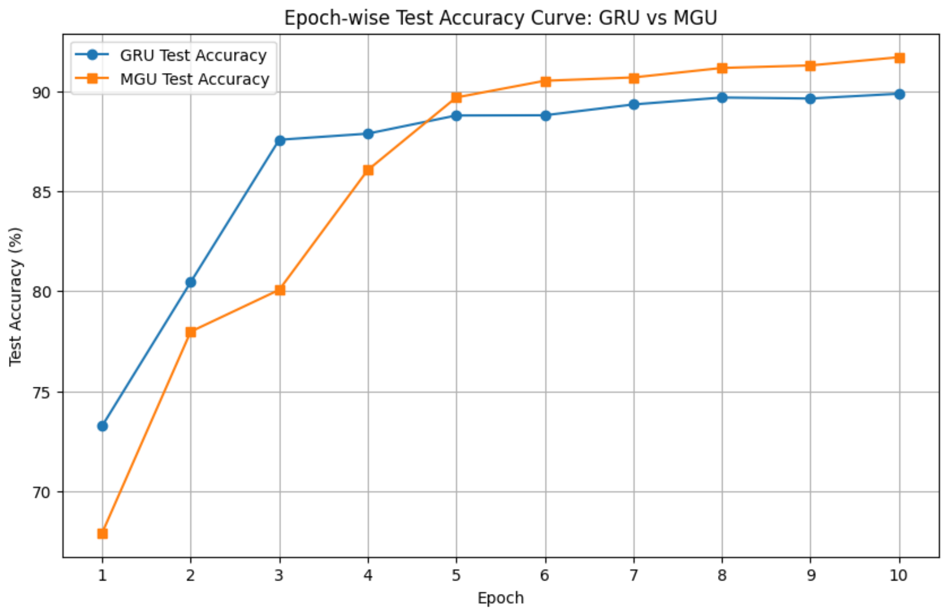

Figure 3: GRU vs MGU test accuracies

1.3 My findings

When comparing the GRU and MGU models trained on the notMNIST small dataset, I observed several important differences in their behavior:

First, both models achieved high training and test accuracies. However, the MGU consistently outperformed the GRU in terms of test accuracy. The average test accuracy for the MGU was 91.69%, compared to 89.86% for the GRU. This indicates that the MGU had slightly better generalization to unseen data.

Looking at the training dynamics, the MGU converged faster and had a smoother loss curve compared to the GRU. In the GRU's case, the training loss sometimes fluctuated or decreased more slowly, especially in the early epochs. In contrast, the MGU reached lower loss values more quickly and showed less overfitting between the training and test losses.

One possible reason for this behavior is the simplified gating mechanism of the MGU. By using only a forget gate instead of both update and reset gates, the MGU may reduce the risk of overfitting and make optimization easier, especially on relatively simple classification tasks like notMNIST. The GRU, while powerful, may introduce more complexity than necessary for this dataset.

Additionally, the classification error for MGU (8.31%) was significantly lower than that of GRU (10.14%). A 2% difference, while seemingly small, is meaningful when evaluating models of similar architectures — especially given that the MGU has fewer trainable parameters and should be computationally more efficient.

Overall, the experimental results confirm that minimal gating mechanisms like the MGU can achieve comparable or even superior performance to more complex units like the GRU on certain tasks, while potentially offering benefits in terms of training speed and model simplicity.