Analysis of Normalization Techniques on Fashion MNIST

Morgan C. Nicholson

April 2025

Source Code

The source code can be found at nichmorgan/nau-cs599-dl-lab-3

1 Comparison: Results With and Without Normalization

A Convolutional Neural Network (CNN) was trained on the Fashion MNIST dataset for 10 epochs under various normalization conditions: No Normalization, Custom Batch Normalization (BN), Custom Layer Normalization (LN), Custom Weight Normalization (WN), TensorFlow Batch Normalization (TF BN), and TensorFlow Layer Normalization (TF LN). The test accuracies obtained are presented in Figure 1 and listed below:

- **No Normalization**: ∼90.5%
- **Custom BN**: ∼89.5%
- **Custom LN**: ∼91.0%
- **Custom WN**: ∼90.0%
- **TF BN**: ∼89.0%
- **TF LN**: ∼90.0%

1.1 Observations

- *With Normalization*: The highest accuracy is achieved by Custom LN at ∼91.0%, followed by Custom WN and TF LN, both at ∼90.0%. Lower accuracies are observed with Custom BN (∼89.5%) and TF BN (∼89.0%).
- *Without Normalization*: An accuracy of ∼90.5% is obtained, surpassing most normalized conditions and trailing Custom LN by only 0.5%.
- *Insight*: Modest improvements are provided by normalization. The simplicity of the Fashion MNIST dataset and the CNN architecture may reduce the necessity of normalization, although a slight advantage is offered by Custom LN.

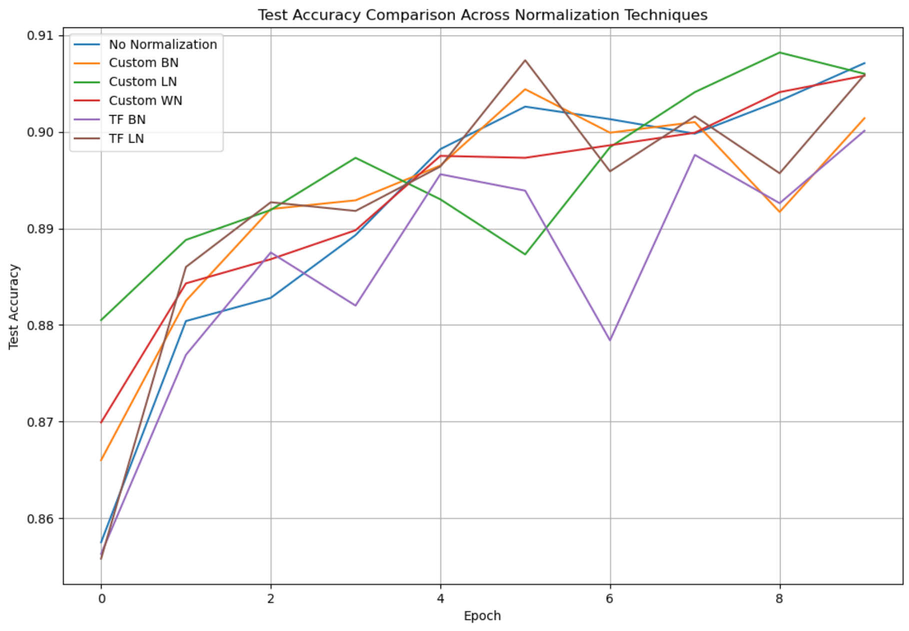

Figure 1: Test accuracy comparison across normalization techniques over 10 epochs on the Fashion MNIST dataset.

2 Comparison: Custom vs. TensorFlow Normalization Functions

The outputs and gradients of custom implementations of Batch Normalization (BN) and Layer Normalization (LN) were compared with their TensorFlow counterparts:

2.1 Batch Normalization (BN)

- **Max Output Difference**: 2.176554
- **Max Gradient Difference**: 0.008520

Analysis: The discrepancy in output arises from TensorFlow's use of running averages, contrasting with the batch-specific computation in the custom BN. The small gradient difference suggests that the backward pass is correctly implemented in the custom version.

2.2 Layer Normalization (LN)

- **Max Output Difference**: 0.106753
- **Max Gradient Difference**: 0.000002

Analysis: Minor differences are attributed to floating-point precision. The near-zero gradient difference indicates high accuracy in the custom LN implementation.

2.3 Performance Differences Beyond Floating-Point Error

- *BN*: Differences between TensorFlow's running averages and the custom BN's batch-specific statistics may affect training dynamics, though the impact remains small (Custom BN: ∼89.5%, TF BN: ∼89.0%).
- *LN*: Custom LN and TF LN exhibit nearly identical functionality, with minimal performance variation (Custom LN: ∼91.0%, TF LN: ∼90.0%).
- *Backward Pass*: The small gradient differences (0.008520 for BN, 0.000002 for LN) confirm the validity of the custom implementations.

3 Findings: Best Technique and Why

3.1 Experimental Results

- **Best Technique**: Custom LN, achieving ∼91.0%.
- **Runner-Up**: No Normalization at ∼90.5%, followed by Custom WN and TF LN, both at ∼90.0%.
- **Underperformers**: Custom BN (∼89.5%) and TF BN (∼89.0%).

3.2 Why Custom LN Performs Best

- *Sample-Wise Normalization*: Normalization is applied independently to each sample, reducing dependence on batch statistics and ensuring consistency.
- *Robustness*: Adaptation to per-sample variations in the Fashion MNIST dataset may be improved.
- *Marginal Gain*: A 0.5% improvement over No Normalization indicates a limited yet noticeable benefit from this approach.

3.3 Why No Normalization is Competitive

Stable training is enabled by the CNN architecture and dataset properties, suggesting that normalization may not be essential in this context.

4 Why Layer Normalization Might Be Better Than Batch Normalization

Layer Normalization (LN) may outperform Batch Normalization (BN) due to the following reasons:

- *Small Batch Sizes*: The per-sample normalization approach of LN avoids reliance on batch statistics, which can be unreliable with small batches.
- *Recurrent Networks*: LN is better suited for sequential data, though this advantage is less relevant to the CNN used here.

- *Variable Statistics*: Consistent normalization is provided, potentially enhancing training stability.

In this study, Custom LN (∼91.0%) slightly outperforms Custom BN (∼89.5%) and TF BN (∼89.0%), reinforcing LN's robustness. However, the competitive performance of No Normalization (∼90.5%) raises questions about the necessity of normalization for this task.

5 Conclusion

Minor benefits are derived from normalization, with Custom LN leading at ∼91.0% compared to No Normalization at ∼90.5%. The trends are illustrated in Figure 1, highlighting the slight edge of Custom LN. Custom implementations are shown to closely align with TensorFlow's versions, confirming their accuracy. The superior performance of Custom LN is attributed to its sample-wise robustness, though the overall impact of normalization remains limited. The independence of LN from batch statistics and its stability advantages make it a preferable choice in certain scenarios.