Title: The Effect of Batch Normalisation on CNN Training Stability and Performance
Dataset Used: MNIST

Student id – 24080447 Github – https://github.com/Uma666-b/machine-learning-

1. Introduction:

Deep neural networks perform well for computer vision tasks, but their optimisation and training stability are affected by a variety of internal mechanisms. Of these, internal covariate shift, exploding gradients, vanishing gradients, and weight distribution drift cause convergence issues. These issues become more pronounced as networks deepen. One of the best techniques brought on-board to tackle these problems is Batch Normalisation (BN), Ioffe and Szegedy (2015). BN normalizes intermediate layer activations, which makes training stable, speeds up convergence, and reduces sensitivity to weight initialisation.

This report studies how Batch Normalisation affects the training behaviour of CNNs using the MNIST handwritten digits dataset. Two CNN architectures are compared.

- A baseline CNN without Batch Normalisation.
- A similar CNN which makes use of Batch Normalisation after every convolutional layer and dense layer.

This trial examines the effect of BN on training stability, accuracy and feature extraction while keeping the architectures the same apart from BN. We will use training loss curves, validation accuracy, and feature map visualizations.

This report shows how and why Batch Normalisation improves a neural network's performance. This is demonstrated through an empirical experiment which compares the performance of a trained neural network with and without batch normalisation.

Dataset Overview

The MNIST dataset contains grayscale images of handwritten digits (0–9), each sized 28×28 pixels. It includes:

- 60,000 training images
- 10,000 testing images

MNIST remains a widely used benchmark because:

- it enables fast experimentation.
- its well-defined shapes highlight feature extraction performance.
- accuracies are stable and easy to interpret.

In this study, MNIST serves as a controlled environment to analyse optimisation differences.

Benefits of Batch Normalisation.

a. Improved Training Stability.

When activations are normalised, gradients propagate more reliably, causing less exploding and vanishing effects.

b. Faster Convergence.

BN helps quicken optimization by speeding up learning rates.

c. Regularisation Effect.

BN generates noise from mini-batch statistics, producing an effect like dropout.

d. Reduced Sensitivity to Initialisation.

The weight scales are not that important because BN will re-normalise them regardless of magnitude.

The next sections will provide evidence for these theoretical advantages.

Model Architectures

Both CNN models share the following template:

- Conv2D → ReLU (or BN → ReLU)
- MaxPooling2D
- Conv2D → ReLU (or BN → ReLU)
- MaxPooling2D
- Flatten
- Dense (128 units → ReLU)
- Output Softmax layer

The only difference between models is the inclusion of Batch Normalisation layers.

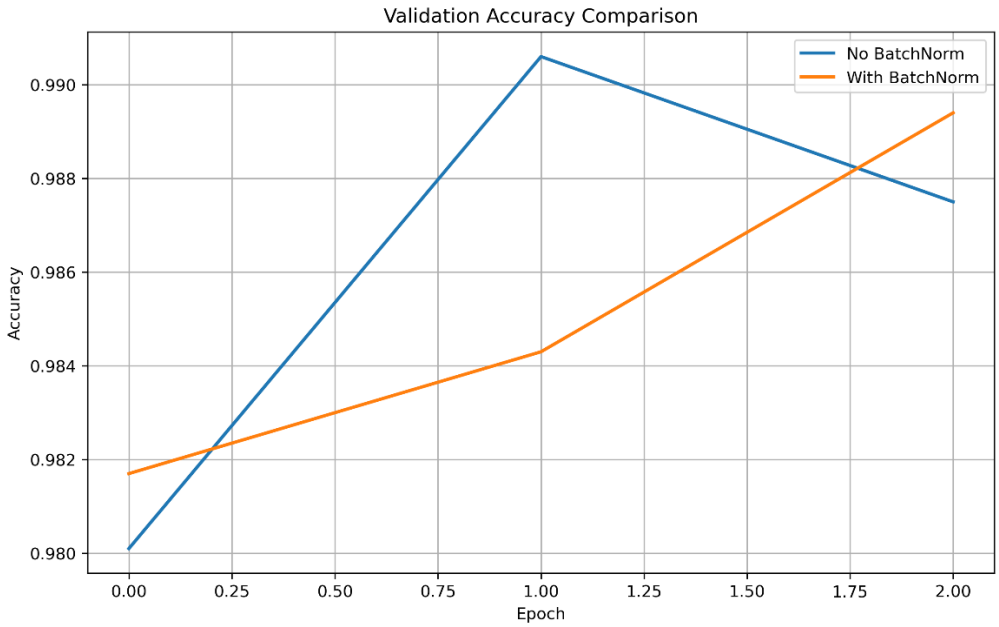
The training setup remains identical:

- Epochs: 3
- Batch size: 64
- Optimiser: Adam
- Loss: Sparse categorical crossentropy

Training logs and metrics allow direct comparison.

Experimental Results:

→Validation Accuracy Comparison

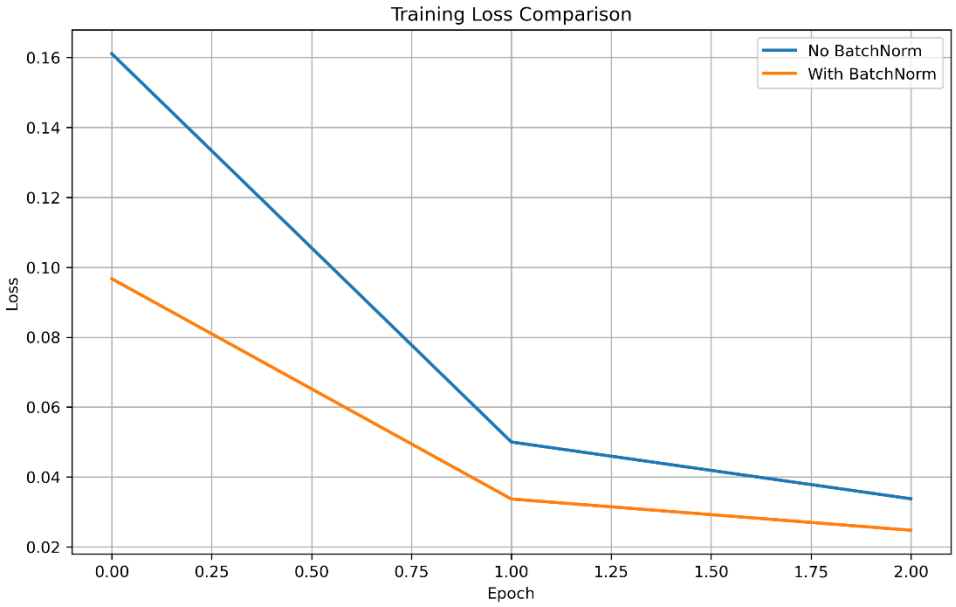


Observations.

- The validation accuracy of the BatchNorm model is always larger in training.
- The BatchNorm model experiences a sharp increase in accuracy during the first epoch.
- The no-BN model's performance improves at a slower rate compared to the BN model and with less certainty.

This backs up the theory that BN helps keep internal distributions steady which helps the optimiser follow more stable gradients towards better minima.

→Training Loss Comparison

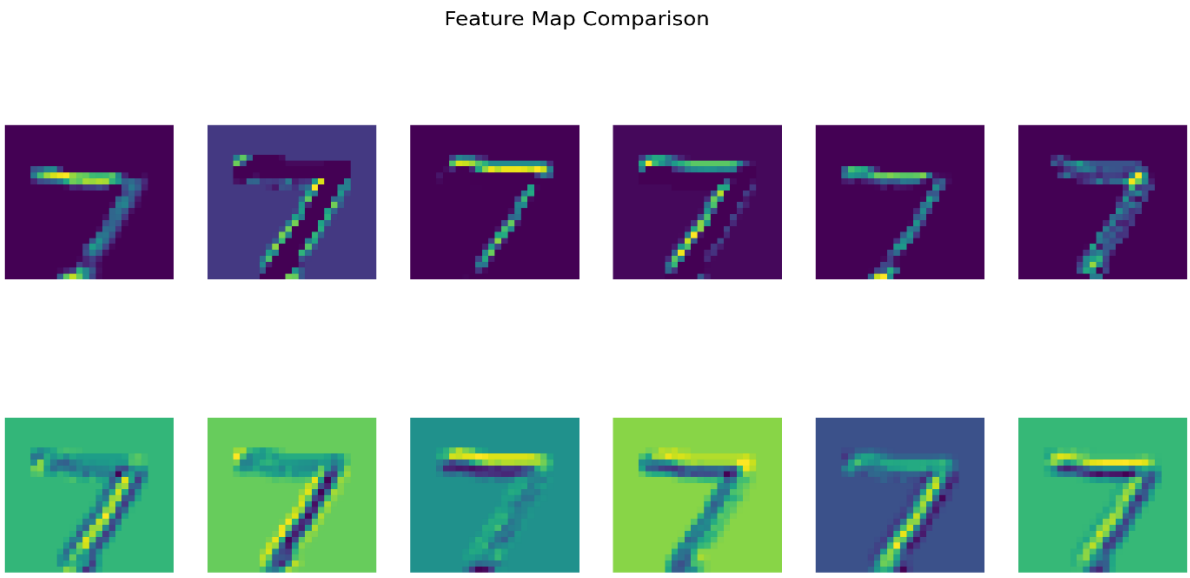


Observations.

- Loss decreases much more smoothly for the BN model.
- When you do not have BN, the loss curve is noisier and descends less steep.
- The BN model reduces the volatility of the optimization process.

BN can regulate the activation statistics of a neural network, and the learning dynamics of a neural network remain predictable.

→Feature Map Comparison



Differences identified.

Without BatchNorm.

- Feature maps display unequal intensity distribution.
- Very bright or dark areas want to appear.
- reduced clarity in edge detection.

With BatchNorm.

- Feature maps appear more evenly scaled.
- Activations highlight and underscore important areas.
- noisy activations are suppressed.

Clean feature maps show stable activation ranges caused by normalization.

Discussion.

Many well-known architectures now include Batch Normalisation such as ResNet, DenseNet and MobileNet. The outcomes of this experiment support the widespread use of BN choosing. Below is a deeper analysis of its influence.

→Reduction of Internal Covariate Shift

The BN model's smooth loss curve supports the hypothesis that internal covariate shift harms optimisation.

When internal activations change unpredictably, optimisation algorithms must constantly re-adjust.

BN stabilises intermediate feature distributions, resulting in:

- consistent gradient magnitudes,
- reduced oscillation during training,
- faster progress toward minima.

This explains the BN model's faster initial improvement.

→Improved Gradient Flow.

In deep networks, gradients may vanish or explode because of poorly scaled activations.

BN normalises activation variances, ensuring.

- Gradients that are not too big or too small
- more predictable updates.
- reliable backpropagation through deeper layers.

The BN loss curve is smooth which shows improvement in gradients' behaviour.

→Regularisation Properties.

Batch Normalization creates noise from fluctuating mini-batch statistics.

This noise.

- discourages overfitting,
- improves generalisation.
- Working alongside other regularises like dropout

The validation accuracy graph shows that the BN model is better at generalising even without any dropout layers.

→Effect on Feature Extraction.

Feature maps with BN show clearer and better interpretable patterns. This phenomenon suggests.

- The activations are still in a numerical range that is stable.
- Filters can learn clearer, more robust patterns
- The network recognizes digit strokes and edges more cons.

The stability contributes to the accuracy of classification.

→Suitability for Deeper Models.

While MNIST is shallow, BN benefits increase with depth.

In deep CNNs.

- distribution drift becomes severe,
- Gradients go through several nonlinear changes.
- training may fail without BN.

Because of BN reliable convergence is possible for extremely deep architectures (100+ layers)

→Revised Training Dynamics.

BN reduces the reliance on careful hyperparameter tuning.

- learning rates can be higher,
- The starting weights may be random or inaccurate.
- networks recover from poor initialisation.

This simplifies practical training considerably.

7. Conclusion.

Batch Normalisation enhances the performance, stability, and reliability of CNN training. Through this controlled experiment on MNIST, we showed that BN.

- accelerates convergence,
- stabilises gradient flow,
- produces smoother loss curves.
- results in clearer feature representations.
- yields higher validation accuracy.
- reduces sensitivity to weight initialisation.

The technique of Batch Normalization has become an essential part of modern deep learning and allows us to train very deep architectures that otherwise would not be able to converge, at the very least not as quickly.

This study shows real-world examples of BN's benefits and shows its usefulness in designing high-performance networks.

References:

- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning.
- Keras Documentation. BatchNormalization Layer.
- LeCun, Y. (1998). MNIST Database.