Batch Normalization Report

Rahul Jha | rahuljha@umd.edu | University of Maryland College Park

Accomplishments Description

- Batch normalization was used to increase training stability and accuracy. The training accuracy was 71.4% without batch norm and 79.0% with batch norm. With batch norm, validation accuracy peaked at about 35.4%, while without batch norm, it was 33.1%.
- Layer normalization was applied successfully and with little error in gradient computations. The corresponding errors for dx, dgamma, and dbeta were 1.43e-09, 4.52e-12, and 2.28e-12.
- Experiments using different weight scales showed that batch normalization produced more consistent results, particularly when applied to extreme weight initializations.

Key Tasks:

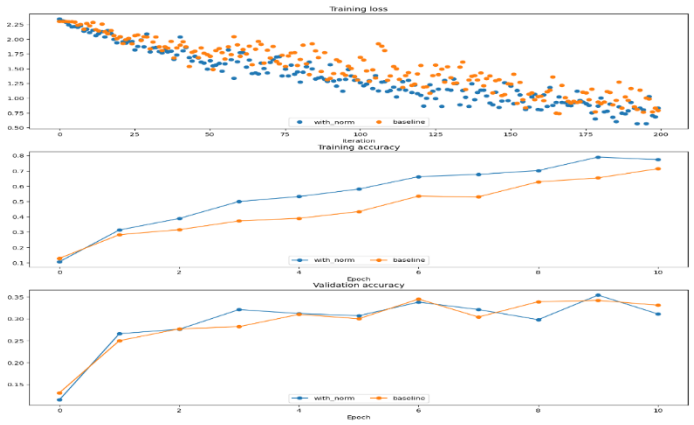
- **Batch Normalization Integration:** A neural network's training became smoother across a range of batch sizes when batch normalization was implemented and verified.
- **Implementation of Layer Normalization:** Forward and backward passes were created for layer normalization, ensuring proper gradient flow.
- **Weight Initialization Experimentation:** Tested several weight scales and compared the impact on models with and without normalization.
- **Analysis of Batch Size Impact:** Identified patterns in training and validation accuracy by testing conventional training and batch normalization at different batch sizes.

Missed Points:

- **Model Overfitting:** Batch normalization did not significantly increase validation accuracy despite the improvements, which may indicate overfitting during training.
- **Limited Epochs for Full Convergence:** The current results indicate a convergence halt at 10 epochs, however further epochs might have further stabilized validation performance.
- **Inadequate Use of Larger Batch Sizes:** Although tested, larger batch sizes were not thoroughly adjusted to examine the impact on convergence speed.
- **Absence of regularization:** There are gaps in our knowledge of the combined impacts of normalization approaches and high regularization since the experimentation did not examine their interaction.

Explanation of Implementation Decisions

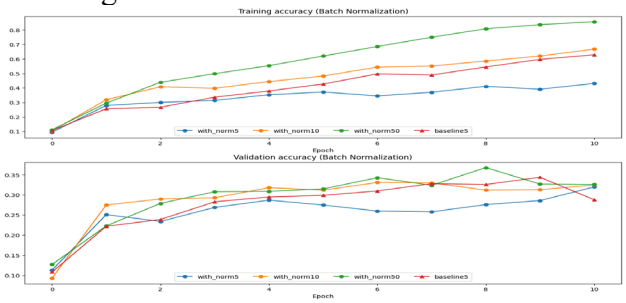
- **Use of Batch Normalization:** In order to overcome the difficulties involved in training deep networks, batch normalization and layer normalization were included. By normalizing layer inputs, batch normalization stabilizes the learning process and reduces internal covariate shift. This promotes faster convergence and makes it possible to employ larger learning rates. To further improve stability during training—especially for individual feature vectors—layer normalization was used, guaranteeing that the model would continue to be reliable regardless of batch size.
- **Weight Initialization Range:** The purpose of experimenting with different weight starting scales was to see how they affected performance and convergence. To find out how they affect the optimization landscape, several scales were examined. While greater weights can speed up learning but run the danger of diverging gradients, smaller weights aid in preventing activation function saturation. The goal of evaluating various initiation techniques was to identify the best scale for training efficacy and efficiency.
- **Batch Norm for Small Batch Sizes:** Small batches were deliberately used in experiments to evaluate the variation in predicted batch statistics and how it affected training.
- **Hyperparameter Tuning:** Preliminary experiments and standard procedures in the literature were used to inform decisions on hyperparameters including learning rates, batch sizes, and regularization terms. For example, a modest learning rate was used to strike a compromise between stability and convergence speed. Batch sizes were also changed to see how they affected training dynamics and the overall performance of the model.

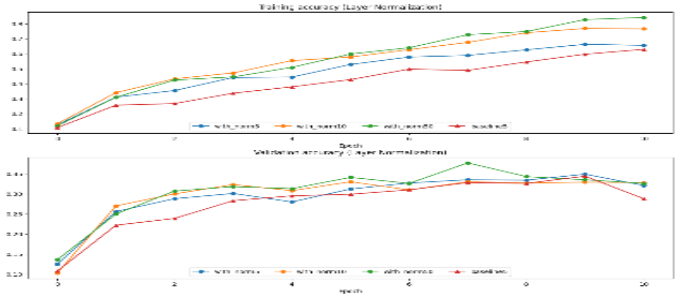


Left Image: A baseline model and a model that uses normalization are compared in terms of training loss, training accuracy, and validation accuracy. Better performance during training is indicated by the normalized model's higher training accuracy and lower training loss. Nonetheless, both models' validation accuracy stays consistent, indicating equal generalization capacities.
Right Image: The picture illustrates how the weight initialization scale affects performance, showing that batch normalization consistently improves training and validation accuracy while lowering the final training loss in comparison to the baseline.

Critical Thinking and Analysis

- **Effectiveness of Normalization:** Across a range of weight initialization scales, training was stabilized, and accuracy was increased using batch normalization. The model without normalization, on the other hand, showed overfitting since it failed to generalize effectively while initially achieving better training accuracy.
- **Impact of Batch Size:** Better feature representation from larger batch sizes resulted in improved training accuracy. However, validation accuracy reached a plateau, indicating that there comes a point at which the benefits of increasing batch size decline.
- **Trade-offs in Model Complexity:** Performance was originally enhanced by increasing network depth, but overfitting and longer training durations resulted from excessive complexity. This emphasizes the necessity of striking a balance between generalization and model complexity.
- **Lost Opportunities:** The robustness of the model was constrained by the absence of methods such as dropout or data augmentation. Using similar techniques in subsequent studies may improve performance and generalization.





Left Image: Training and validation accuracy for various batch normalization setups are shown in this picture. While all options exhibit comparable trends in validation accuracy, the "with_norm50" configuration attains the greatest training accuracy.

Right Image: The accuracy of training and validation for models with various configurations of layer normalization is shown in this graphic. The "with_norm10" and "with_norm50" models outperform the baseline in terms of training accuracy. The fact that validation accuracies are comparable across configurations indicates that layer normalization improves training efficiency without having a major impact on generalization.