**Convolution Neural Network Report**

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

## Accomplishments Description

- Forward and backward passes of spatial batch normalization were implemented, and the output mean was close to zero and the standard deviation was close to one, indicating proper normalizing.
- Channels were divided into groups and each was normalized individually as part of the implementation of spatial group normalization.
- **Spatial batch normalization**: dx error: 2.786648e-07, d$\gamma$ error: 7.097482e-12, dβ error: 3.275609e-12
- **Spatial group normalization**: dx error: 7.413110e-08, d$\gamma$ error: 9.468196e-12, dβ error: 3.354494e-12
- Tested **non-trivial $\gamma$ and β** scaling for spatial batch normalization with custom parameters resulting in correct shifts and scaling factors: Means: [6, 7, 8] Standard deviations: [2.999999, 3.999999, 4.999998]

**Key Tasks:**

**Spatial Batch Normalization**

- **Forward Pass:** Reshaped input from (N, C, H, W) to (N*H*W, C); normalization results showed means of ([6.1895e-16, 5.9952e-16, -1.2212e-16]) and standard deviations of ([0.99999962, 0.99999951, 0.9999996]). After applying ($\gamma$ = [3, 4, 5]) and (β = [6, 7, 8]), output means were ([6, 7, 8]) with standard deviations of ([2.999999, 3.999999, 4.999998]).
- **Backward Pass:** Implemented gradient propagation with reshaped tensors; gradient check results yielded (dx) error: (2.7866e-07), (d$\gamma$) error: (7.0975e-12), (dβ) error: (3.2756e-12).

**Spatial Group Normalization**

- **Forward Pass:** Reshaped input to ((N, G, C//G, H, W)) for group-wise normalization; means were ([-2.1464e-16, 5.2551e-16, 2.6553e-16, -3.3862e-16]) and standard deviations of ([0.99999963, 0.99999948, 0.99999973, 0.99999968]).
- **Backward Pass:** Conducted backpropagation for input, $\gamma$, and β; gradient check results showed (dx) error: (7.4131e-08), (d$\gamma$) error: (9.4682e-12), (dβ) error: (3.3545e-12).

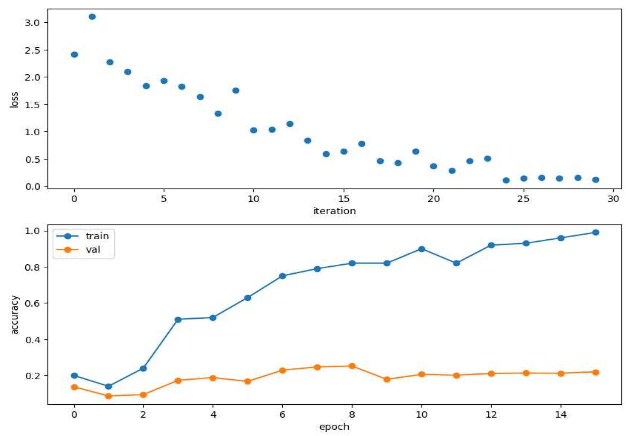
**Missed Points:**

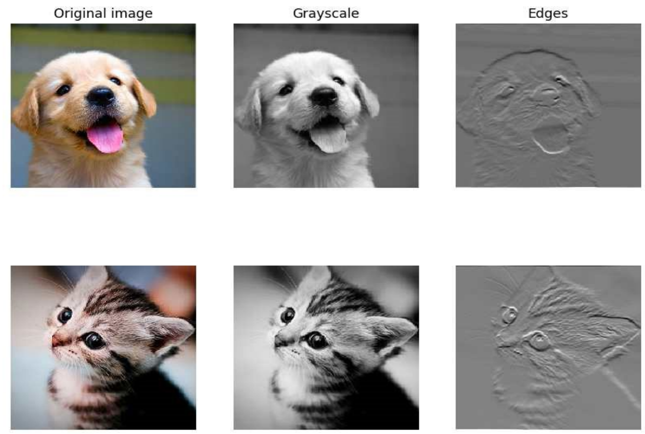
- **Handling Overfitting:** Despite achieving high training accuracy, overfitting became apparent with low validation accuracy. Future work could involve applying regularization techniques or data augmentation to mitigate overfitting.
- **Testing on Bigger Datasets:** A small sample was used for the validation. Testing the solution on bigger datasets in order to evaluate scalability and performance under various circumstances may be part of future development.
- **Performance Profiling:** Especially in the setting of large neural networks, further profiling of the forward and backward passes may highlight areas for performance enhancement.
- **Integration with Other Layers:** Examining how to integrate the neural network design with other layers may improve the overall performance of the model, especially in deeper networks where normalization effects compound.
- **Regularization Techniques**: Model generalization may be enhanced by examining the effects of using regularization techniques (such as dropout) in addition to normalization procedures.

## Explanation of Implementation Decisions

- **Reshaping for Batch Normalization:** To take use of the vanilla batch normalization function that was already in place, the input tensor was reshaped from shape $(N,C,H,W)$ (N,C,H,W) to shape $(N{\cdot}H{\cdot}W,C)$ (N·H·W,C) in order to apply spatial batch normalization. Each channel may be separately normalized while preserving the input's spatial structure thanks to this method's more effective mean and variance calculation across spatial dimensions.

- **Layer-wise Operations:** Per-channel operations were carried out during both forward and backward passes. This choice is crucial for maintaining the statistical independence of each channel, which is necessary for efficient normalization. Reshaping makes ensuring that the shifting and scaling parameters ($\gamma$ and β) are applied appropriately throughout the relevant dimensions.
- **Group Normalization Design:** To support the designated number of groups G, the input tensor was modified for group normalization. This choice eliminates problems associated with batch normalization's dependence on large batch statistics and permits the independent normalizing of groups, which is especially advantageous in situations where batch sizes are small or change.




**Left Image:** The model is successfully learning, as seen by the first graph, which shows a decreasing loss over iterations. The second graph, which displays training and validation accuracy over epochs, indicates possible overfitting since training accuracy increases dramatically while validation accuracy stays mostly constant.

**Right Image:** The many image processing phases are depicted in this collection of pictures. The original color photos of a puppy and a kitten are displayed in the first column. These pictures are converted to grayscale in the second column, which eliminates color information. The third column draws attention to the pictures' edges, highlighting their forms and outlines.

## Critical Thinking and Analysis

- One design decision that improves code modularity and performance is the use of reshaping to make spatial batch normalization compatible with regular batch normalization. This minimizes code duplication and facilitates maintenance.
  In situations when batch sizes are lower or more variable, group normalization performs better than batch normalization since it was successfully developed to function without being affected by batch size.
- **Trade-offs in Model Complexity**:
  - Batch normalization typically requires a larger batch size for better statistics. When batch size is constrained, group normalization is a suitable trade-off since it can function better with smaller batches.
  - Although group normalization offers greater flexibility, the requirement to divide channels into groups introduces an extra level of computational complexity. Although there is a tiny increase in memory footprint, the performance gains in smaller batch regimes make the trade-off justifiable.
- **Lost Opportunities**
  - Hyperparameter Optimization: Standard settings for $\gamma$γ and $\beta$β were the main emphasis. Better convergence and increased accuracy may have resulted by investigating a larger range of hyperparameters or via automated hyperparameter adjustment.
  - Data Augmentation: Using sophisticated data augmentation methods during training could have reduced overfitting, increased validation accuracy, and strengthened the model's resilience.
  - Dropout Integration: By adding dropout layers to the network, more regularization might have been achieved, reducing overfitting and enhancing generalization on unknown input.