EE 228 HW#3 - Data Augmentation

Name: Shaorong Zhang
SID: 862394226

May 20, 2023

1. (3 pts) Train your Resnet model without augmentation and report the results.
Solution:
Final test accuracy is 75.36%.

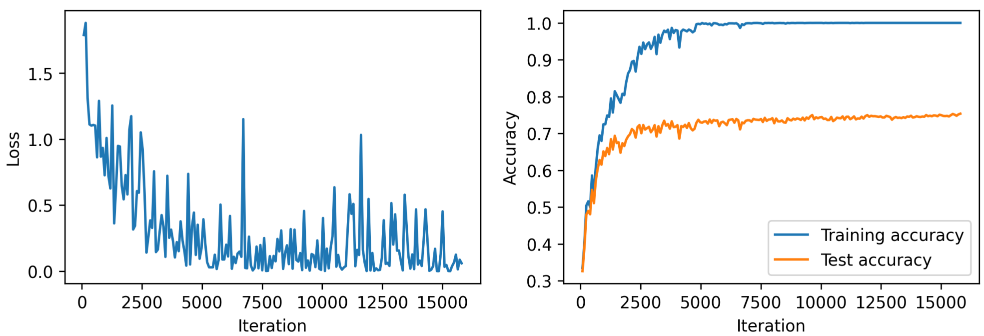

Figure 1: Training loss, training and test accuracy without augmentation

2. (4 pts) Mixup augmentation is based on the paper https://arxiv.org/pdf/1710.09412.pdf. As the name suggests, it mixes a pair of training examples (both inputs and labels). Given a pair of training example $(x_1, y_1), (x_2, y_2)$, we obtain the augmented training example (x, y) via

$$x = \lambda x_1 + (1 - \lambda)x_2 \quad y = \lambda y_1 + (1 - \lambda)y_2$$

TODO: Implement mixup and report the results for $\alpha = 0.2$ and $\alpha = 0.4$. Note that, in each minibatch, all training examples should have mixup transformation before gradient calculation (e.g. from original minibatch obtain a new minibatch by mixing random pairs of training examples).
Solution:
As $\alpha = 0.2$, final test accuracy is 77.64%, as $\alpha = 0.4$, final test accuracy is 79.33%

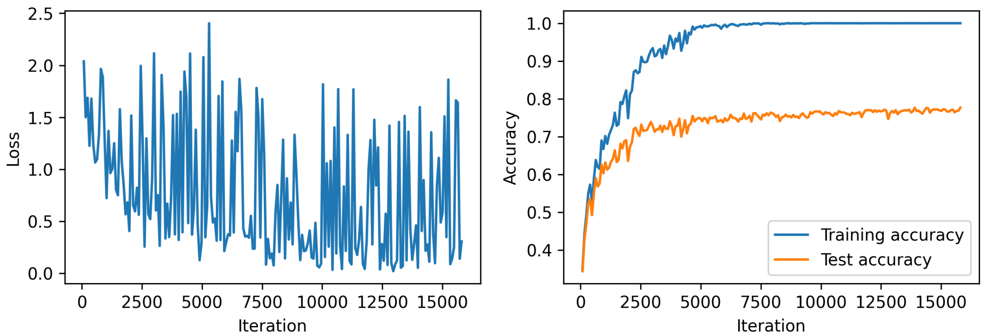

Figure 2: Training loss, training and test accuracy using Mixup augmentation with $\alpha = 0.2$

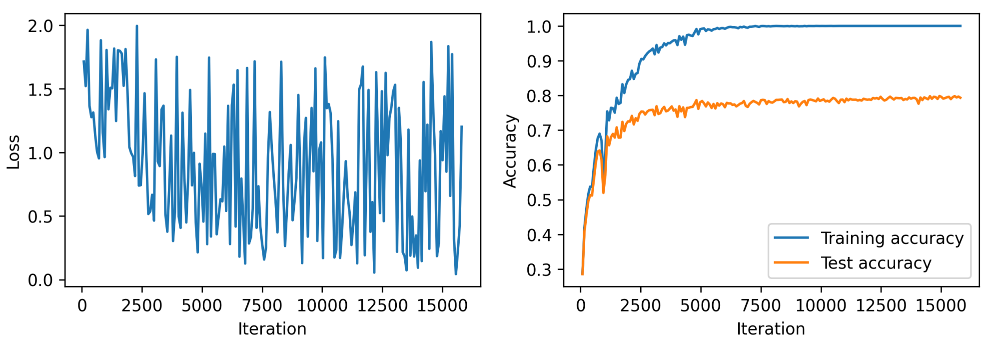

Figure 3: Training loss, training and test accuracy using Mixup augmentation with $\alpha = 0.4$

3. (4 pts) Cutout augmentation is based on the paper https://arxiv.org/pdf/1708.04552.pdf. For each training image with 50% probability you keep the image intact. With probability, select a random pixel which serves as the center of your cutout mask. Then, set the square mask of size $K \times K$ pixels around this center pixel to be zero. Note that part of the mask is allowed to be outside of the image. For visualization, see Figure 1 of the paper.

TODO: Implement and use cutout augmentation with $K = 16$ and report the results.

Solution:

Final test accuracy is 76.63%.

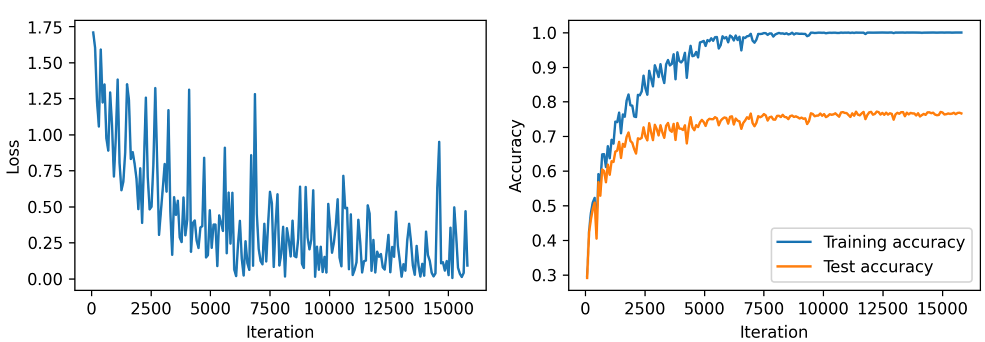

Figure 4: Training loss, training and test accuracy using Cutout augmentation

4. (4 pts) Standard augmentation applies horizontal flip and random shifts. See the website https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-ne for illustrations. Given an input image, first you shift it left-right and up-down as follows. Pick two independent integers k_1, k_2 uniformly between $[-K, K]$ range. Move image upwards by k_1 and rightwards by k_2 pixels (negative value means downwards and leftwards). Zero pad the missing pixels. After this random shift, with 50% probability, apply a horizontal flip on the image.

TODO: Implement standard augmentation with $K = 4$ and report the results.

Solution:

Final test accuracy is 78.86%.

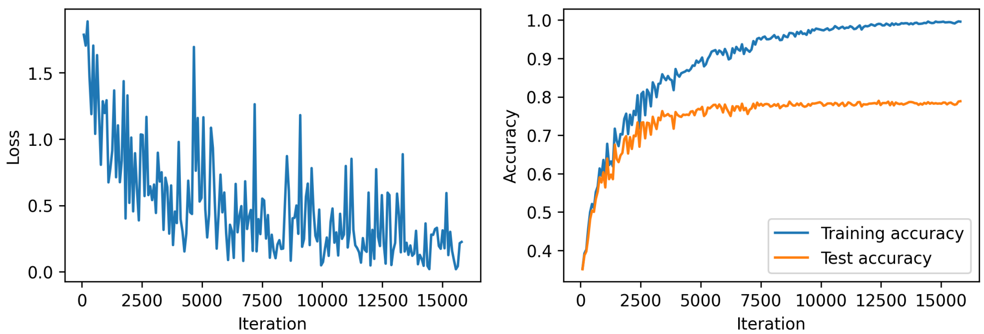

Figure 5: Training loss, training and test accuracy using Cutout augmentation

5. (3 pts) Combine all augmentations together. First apply standard and cutout augmentations on the training images and then apply mixup to blend them. For mixup, use the parameter α that has higher test accuracy. Report the results. Does combining improve things further?

Solution:

Final test accuracy is 81.73%, combining significantly improved the final test accuracy.

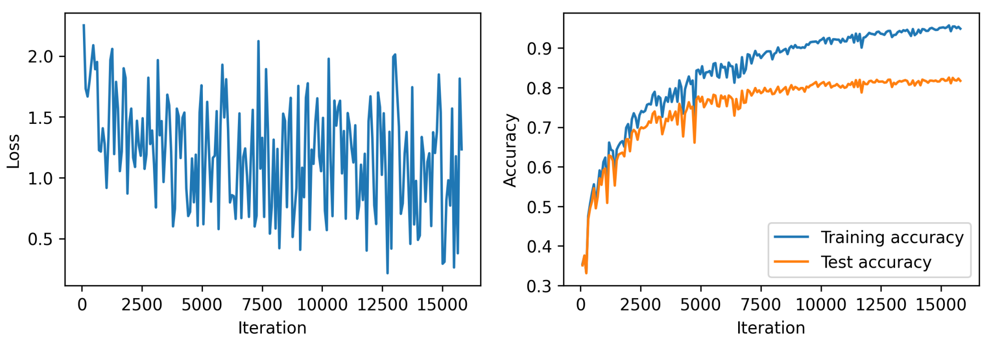

Figure 6: Training loss, training and test accuracy using combined augmentation

6. (2 pts) Comment on the role of data augmentation. How does it affect test accuracy, train accuracy and the convergence of optimization? Is test accuracy higher? Does training loss converge faster?

Solution:

As demonstrated in Table 1, augmentation has proven to be effective in increasing the final test accuracy. However, it is important to note that augmentation often results in lower training accuracy. This discrepancy can be attributed to the fact that the model without augmentation tends to overfit quickly due to the limited number of training examples available.

By introducing augmentation techniques, we can alleviate the issue of overfitting by injecting randomness into the training dataset. Nevertheless, it is worth mentioning that overfitting can still occur if the model's capacity and the number of training iterations are sufficient. Nonetheless, through the use of augmentation, we observe a slower convergence in the optimization process. Despite this, the likelihood of overfitting is significantly reduced, resulting in a decrease in training accuracy while achieving higher test accuracy.

Table 1: Final training and test accuracy of different methods

Methods	Final training accuracy	Final test accuracy
Without augmentation	100.00%	75.36%
Mixup augmentation with $\alpha = 0.2$	100.00%	77.64%
Mixup augmentation with $\alpha = 0.4$	100.00%	79.33%
Cutout augmentation	100.00%	76.63%
Standard augmentation	99.58%	78.86%
Combined augmentation	94.91%	81.73%