EMBEDDED MACHINE LEARNING

Exercise 4

Group 6
Malte Bauer, Mielke Max, Maximilian Burr, Sergej Bespalov
May 21, 2024

Contents

1 Exercise 4.1

We implemented the VGG11 convolutional network, a smaller version of the VGG architecture designed for image classification. VGG11 utilizes a series of convolutional layers with 3x3 filters (Conv3-64), max pooling layers (Maxpool) to reduce spatial dimensions, and fully connected layers (FC-4096) for classification. We adapted the final layer for the CIFAR-10 dataset and trained the network for 30 epochs.

Training on a GPU was significantly faster than on a CPU, with one epoch taking roughly 600 seconds on CPU compared to 20 seconds on GPU displayed in Figure 1. This difference is due to the increased complexity of VGG11 compared to simpler MLPs, as convolutional operations are computationally intensive and well-suited for GPU parallelization.

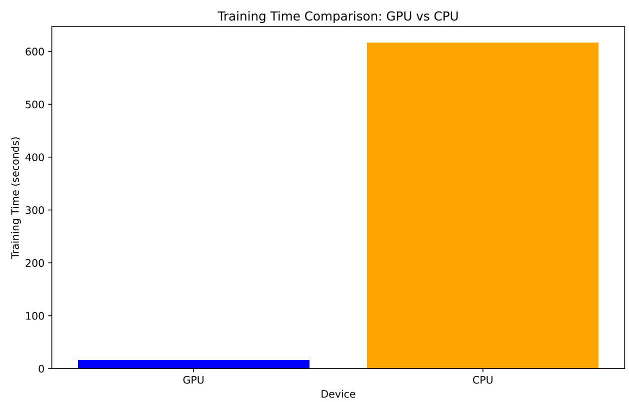

Figure 1: Comparison of training time per epoch between CPU and GPU

After approximately 10 epochs, the test loss plateaued and began increasing, suggesting overfitting as shown in Figure 2. To address this, we introduced dropout layers before each ReLU activation, randomly deactivating neurons during training to promote feature diversity. We experimented with dropout probabilities between 0.1 and 0.9, and the results in 3 show that dropout can improve test accuracy compared to the baseline model without dropout, with the optimal probability dependent on the specific dataset and model complexity.

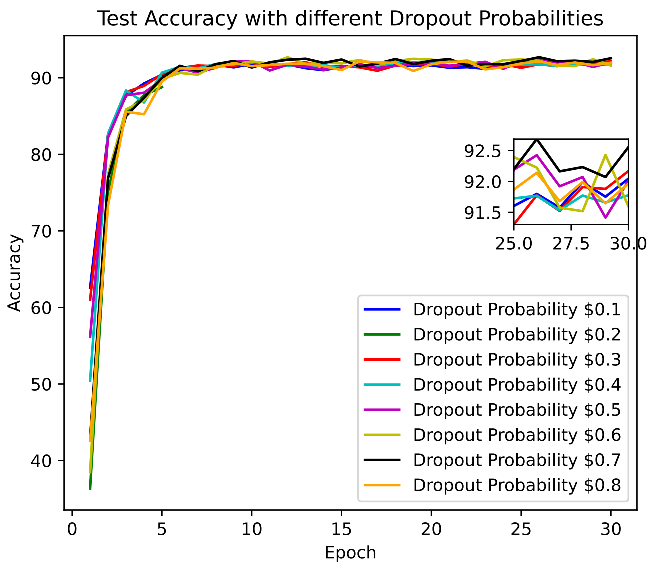

Figure 3: Accuracy with different Dropout Probabilities

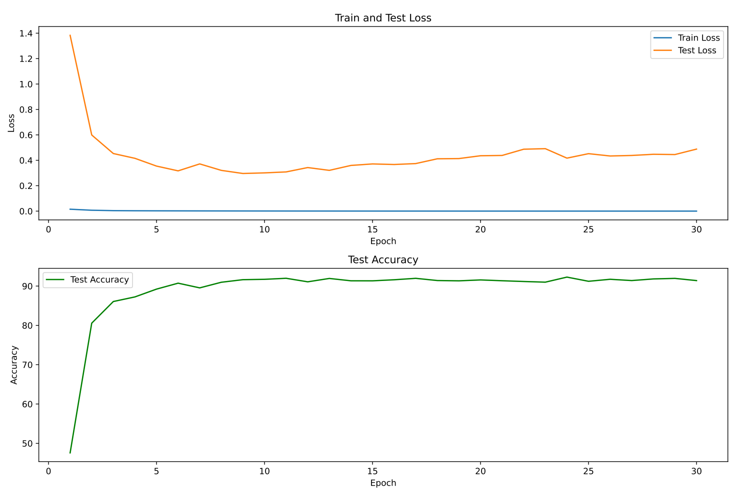

Figure 2: Test accuracy and test loss over the number of epochs

2 Exercise 4.2

We implemented L2 regularization in the VGG11 network by adjusting the weight decay parameter of the ADAM optimizer. We varied the regularization strength (lambda) between 0.001 (strong) and 1e-06 (weak), using multiple measurement points.

The histograms in Figure 3 of the last convolutional layer's weights revealed that stronger L2 regularization (higher lambda) resulted in smaller weight magnitudes. This effect stems from L2 regularization's penalty on large weights, encouraging the network to distribute information more evenly across weights and reduce reliance on a few dominant neurons.

This can enhance generalization and mitigate overfitting.

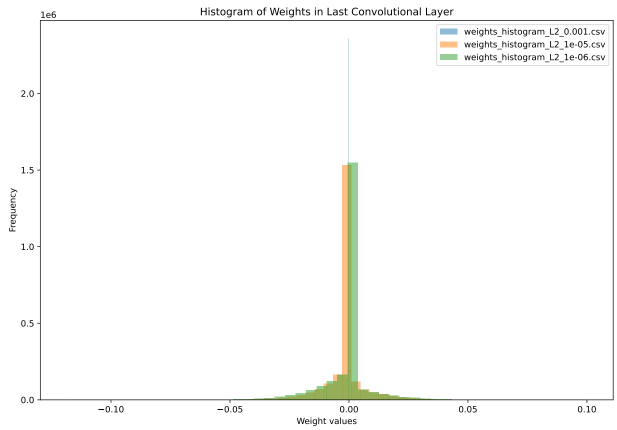

Figure 4: Histogram of weights in the last convolutional layer of VGG11 under varying L2 regularization strengths

The plot of test accuracy over epochs for different L2 regularization strengths is displayed in Figure 4 and demonstrated a clear relationship between regularization and learning behavior. Very strong regularization (lambda = 0.001) severely impaired the network's ability to learn, leading to very low accuracy. Conversely, weak regularization allowed for effective learning, but might not sufficiently prevent overfitting. The optimal lambda value, around 1e-5, seemed to strike a balance, promoting relatively fast learning and achieving the best overall accuracy.

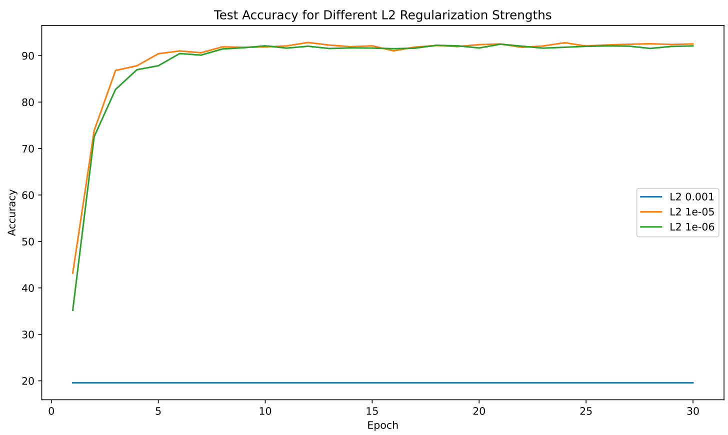

Figure 5: Test accuracy of VGG11 over epochs with different L2 regularization strengths

3 Exercise 4.3

We applied various image data augmentations from torchvision to the CIFAR-10 dataset, including random cropping, horizontal flipping, and a combination of multiple augmentations.

The plot of accuracy over epochs shows in Figure 5 that data augmentation significantly improved model performance. The baseline model without augmentation achieved around 70% accuracy, while random cropping alone yielded a notable increase. The combination of multiple augmentations led to the highest accuracy, surpassing 82%, demonstrating the effectiveness of increasing training data diversity.

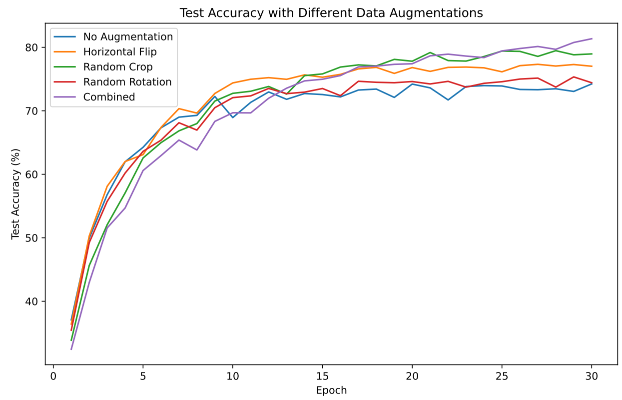

Figure 6: Training accuracy of VGG11 over epochs with different data augmentation techniques

Figure 6 illustrates the training time per epoch for different augmentation techniques. The combined augmentation model took the longest, while random rotation was the fastest. Notably, the baseline model without augmentation had a significantly fluctuations in training time for different epochs.

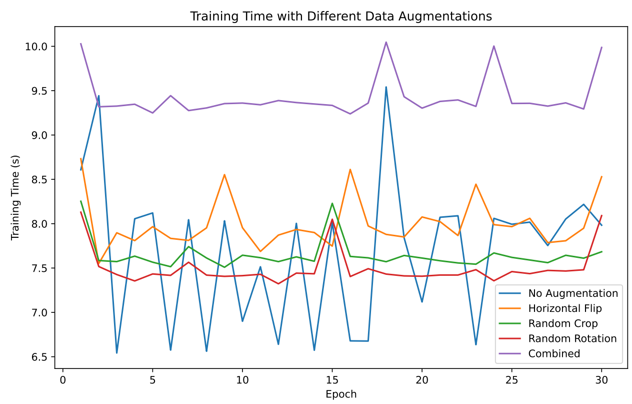

Figure 7: Training time per epoch for VGG11 with various data augmentation techniques

Discussion: Trade-offs Between Regularization Techniques

Horizontal Flip: This simple augmentation improves generalization by preventing overfitting to specific orientations.

Random Crop: Cropping and padding images teaches the model to focus on different parts, helping recognize objects regardless of position.

Random Rotation: Introduces rotational variance, beneficial for datasets with objects in varying orientations.

Combined Augmentations: Generally offers the best performance, combining the benefits of individual augmentations.

Dropout

- **Advantages:** Prevents overfitting by randomly dropping neurons, forcing the model to learn robust features.
- **Disadvantages:** Can slow down training and requires careful tuning of the dropout probability.
- **When to use:** Effective in larger networks prone to overfitting and when neuron co-adaptation is a risk.

L2 Regularization

- **Advantages:** Penalizes large weights, promoting smoother decision boundaries and better generalization.
- **Disadvantages:** Requires tuning of the regularization parameter; excessive regularization can lead to underfitting.
- **When to use:** To control model complexity, prevent excessive weight magnitudes, and improve generalization.

Data Augmentation

- **Advantages:** Increases training data diversity, enhancing generalization and reducing overfitting without collecting new data.
- **Disadvantages:** Can increase training time due to on-the-fly transformation computations.
- **When to use:** Effective with limited training data or when models struggle to generalize. Helps build invariance to flips, crops, rotations, etc.

4 Willingness to present

Exercise 1	Yes
Exercise 2	Yes
Exercise 3	Yes