Sophia Godfrey
Student ID: 801149485
Homework 7

[Intro-To-ML/Homework 7 at main · QueenSophiaLo/Intro-To-ML]

Problem 1

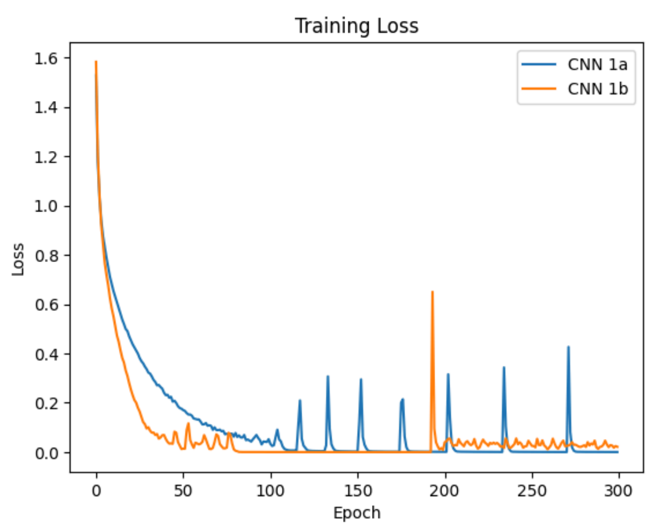
Part a

Problem 1 required building a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset, a widely used benchmark in computer vision containing 60,000 color images (32×32 pixels) across 10 categories such as airplanes, cars, birds, and cats. The fully connected layer at the end of the network was adjusted to match the 10 output classes, and the model was trained for 300 epochs. Throughout training, the training time, training loss, and final evaluation accuracy were recorded.

Part b

For Part 1b, the CNN from Part 1a was extended by adding an additional convolutional layer, followed by a ReLU activation and a pooling layer. The fully connected layer was also adjusted to account for the new intermediate feature dimensions. As in Part 1a, the model was trained for 300 epochs, and the training time, final loss, and evaluation accuracy were recorded.

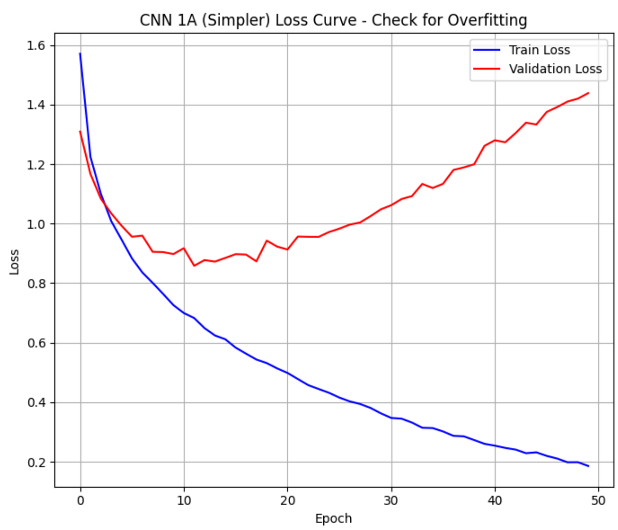
Comparison

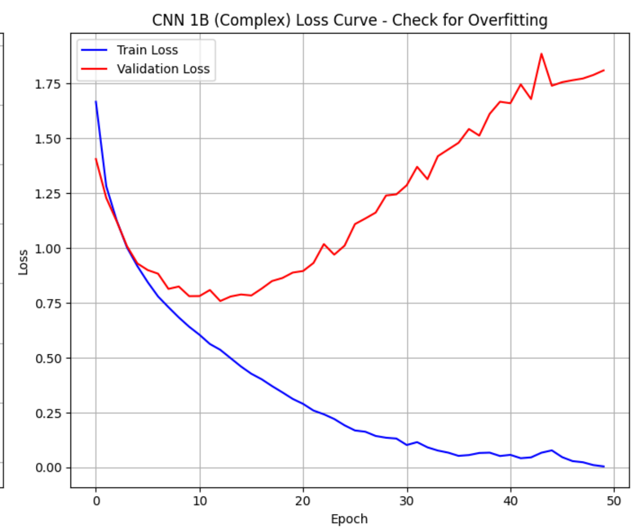



```
CNN 1a - Time: 3414.04s, Accuracy: 68.29%, Params: 60,362
CNN 1b - Time: 3453.85s, Accuracy: 72.88%, Params: 113,738
```

Adding an extra convolutional layer in CNN 1b nearly doubles the number of parameters, resulting in slightly longer training time (~40s more) compared to CNN 1a. However, this additional layer improved the model's feature extraction capability, leading to a higher accuracy of 72.88% versus 68.29% for the baseline CNN. This demonstrates that deeper architectures can better capture complex patterns in images, but the trade-off is increased model size and slightly longer training.

In Homework 2, the fully connected (FC) network was trained using gradient descent with feature scaling and regularization on a tabular housing dataset. FC networks work well for low-dimensional tabular data, but they are not ideal for high-dimensional spatial inputs like CIFAR-10 images because they cannot capture local patterns effectively. Although I do not have the exact model size or training times for the FC network, the difference in data complexity is clear: the Housing.csv dataset contained only a handful of numerical features per example and loaded within seconds on CPU, whereas CIFAR-10 consists of thousands of 32×32 color images and required several minutes to load even with GPU acceleration. Similarly, the FC training in Homework 2 completed within a minute or two on CPU, while training the CNN on CIFAR-10 required roughly an hour on GPU. This contrast highlights how much more computationally demanding image-based models are compared to simple tabular models.





Despite CNN 1B achieving a higher final accuracy, a deeper analysis of the training history reveals severe overfitting in both models caused by the excessive 300-epoch training duration. Both models' ability to generalize peaked extremely early, specifically around Epoch 10-12, after which the Validation Loss began to rise while the Training Loss continued to drop significantly . CNN 1B demonstrated the most extreme case, achieving near-perfect memorization of the training data (Loss ~0.0048) but suffering a massive final generalization loss (~1.81). Continuing training past the optimal performance point (Epoch 10-12) was highly detrimental, as the networks were only learning noise, thereby sacrificing their real-world predictive ability.

Problem 2

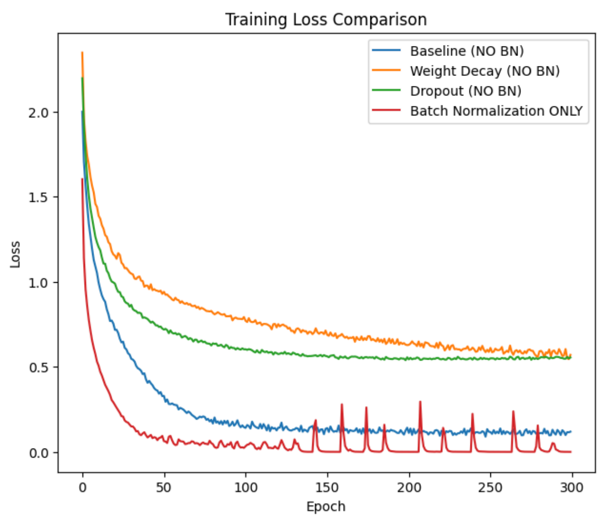
Part a

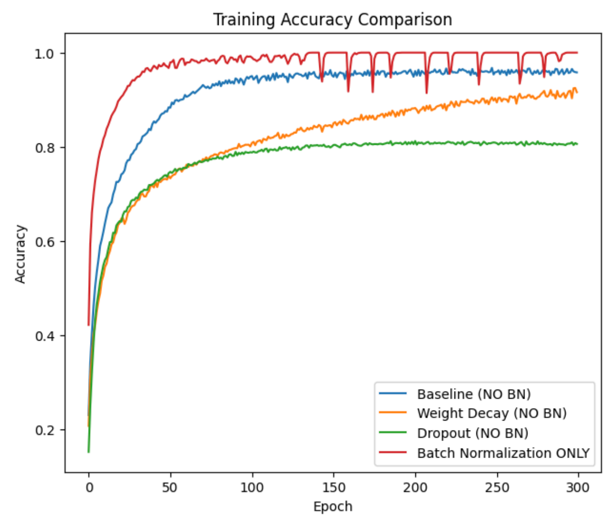
For Problem 2a, a ResNet-based Convolutional Neural Network was built, incorporating skip connections as discussed in lectures, to classify images across all 10 classes in CIFAR-10. The network consisted of 10 residual blocks, forming a ResNet-10 architecture. The model was trained for 300 epochs, and the training time, final training loss, and evaluation accuracy after 300 epochs were recorded.

Part b

For Problem 2b, three additional training experiments were conducted on the ResNet-10 model to evaluate the impact of different regularization techniques. The experiments included: weight decay with a lambda of 0.001, dropout with a probability of 0.3, and batch normalization. For each configuration, the network was trained for 300 epochs, and the training time, final training loss, and evaluation accuracy were recorded and compared across the three approaches.

Comparison





==== Summary of All Experiments ====

Method	Time (s)	Train Loss	Test Acc
Baseline (NO BN/NO Reg)	573.38	0.1191	0.7440
Weight Decay (λ=0.001) (NO BN)	569.45	0.5704	0.7553
Dropout (p=0.3) (NO BN)	723.35	0.5547	0.7642
Batch Normalization ONLY	755.45	0.0004	0.8166

The training loss and accuracy plots for the ResNet-10 experiments highlight the impact of different regularization techniques. The baseline model, without batch normalization or other regularization, shows a moderate training loss that gradually decreases over the 300 epochs and achieves a test accuracy of around 74%. Introducing weight decay increased the training loss slightly, reflecting the regularization effect, but improved test accuracy to approximately 75.5%, indicating better generalization. Dropout also increased training loss, but provided further improvement in test accuracy to roughly 76.4%, demonstrating that stochastic neuron masking helps prevent overfitting. Batch normalization, however, dramatically reduced training loss, reaching near zero early in training, while simultaneously achieving the highest test accuracy of about 81.7%. The loss curves indicate that batch normalization stabilizes and accelerates convergence, while the accuracy plot shows it allows the network to consistently outperform the other methods. Overall, these charts confirm that regularization techniques, especially batch normalization, improve generalization at the cost of slightly longer training times, while the baseline model overfits more quickly and achieves lower accuracy.

The comparison between Problem 1's basic sequential CNNs and Problem 2's ResNet-10 architecture reveals significant differences in efficiency and performance. Despite being a much deeper model, the ResNet baseline (P2) trained six times faster, completing 300 epochs in only 573.38s compared to the basic CNN 1b's 3453.85s, a massive efficiency gain attributed to the ResNet's skip connections . In terms of accuracy and generalization, the ResNet-10 architecture also drastically outperformed its predecessor: while the best basic CNN achieved 72.88% accuracy and suffered severe overfitting (peaking at Epoch 10-12), the ResNet baseline started at 74.40%, and when paired with effective regularization, specifically Batch Normalization, it reached a final maximum accuracy of 81.66%. This nearly 9% absolute increase in test accuracy confirms that modern residual architectures combined with effective regularization are essential for mitigating the severe overfitting observed in the simpler, less stable CNNs of Problem 1.