MP Project 1 Report

Members: Johnny Lin, Matt Wei, Boshen Pan

1. Introduction

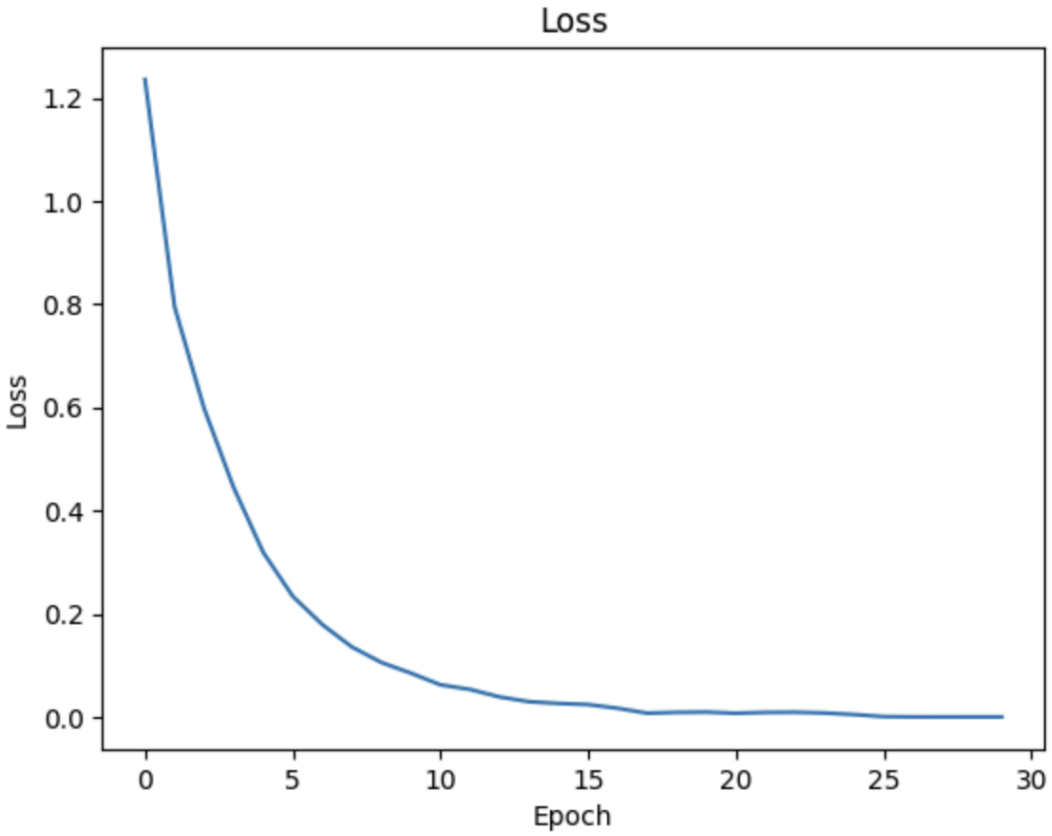
The CIFAR-10 dataset comprises 60,000 32x32 color images across 10 classes. The dataset's division into five training batches and one test batch facilitates extensive training and testing of machine learning models. The DCNN's architecture is designed to learn features from this dataset for accurate image classification effectively.

2. Methodology

2.1 Model Architecture

By importing the PyTorch package, the DCNN architecture used includes 2 convolutional blocks,3 fully connected layers, forward checking as activation functions, etc. The model is trained using “torch.optim.SGD()” as the optimizer with a learning rate of 0.01, “nn.CrossEntropyLoss()” as the loss function, and a default batch size over 30 epochs.

Here is the epoch-wise average loss plot of our DCNN:



2.2 Adversarial Attacks

We implemented two types of adversarial attacks:

- FGSM (Fast Gradient Sign Method): A one-step attack method where adversarial examples are generated by applying the sign of the gradient of the loss function to the input images. This method is fast and efficient for exploring model vulnerabilities.
- PGD (Projected Gradient Descent): An iterative attack method that takes multiple steps with a small step size to find more effective adversarial examples within a specified perturbation budget.

3. Defense Mechanism

To defend against these attacks, we used adversarial training with each attack function, then evaluated the adversarially-trained model with each attack function against the corresponding attack function on test data.

4. Experiments and Results

Experiments were conducted to test the classification accuracy of the DCNN on the original and adversarial CIFAR-10 images. Each attack was evaluated at three noise magnitude levels: low, medium, and high. The results show the percentage drop in accuracy with each attack and the effectiveness of the defense mechanism. In detail, the bigger the value of noise magnitude epsilon, the less accuracy of the classification, and it applies to all cases. The results are shown below:

In the initial setting up of DCNN, we got an accuracy of 82.15% in the test set.

When we implemented the FGSM attack function:

When noise magnitude = 0.001, test accuracy is 10.64%;

When noise magnitude = 0.05, test accuracy is 10.58%;

When noise magnitude = 0.1, test accuracy is 10.53%;

Next, we implemented the PGD attack function instead:

When noise magnitude = 0.01, test accuracy is 10.65%;

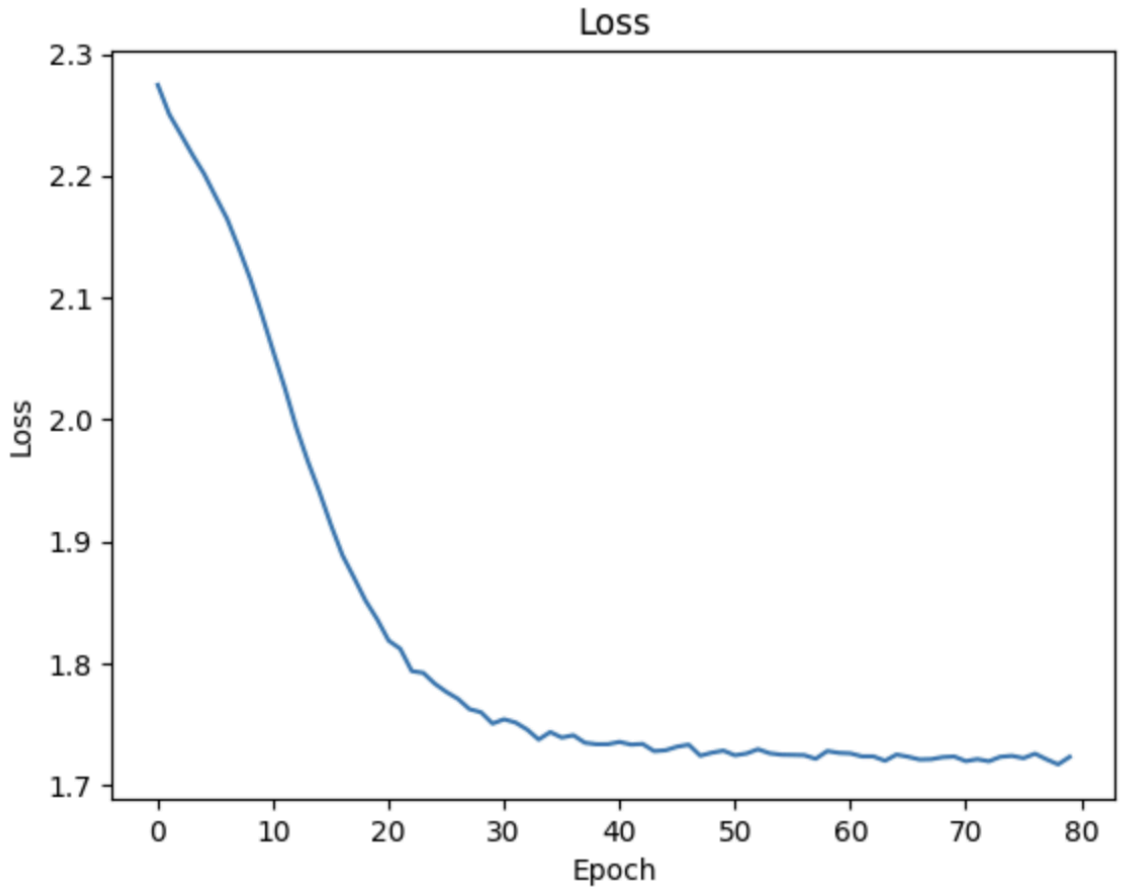
When noise magnitude = 0.05, test accuracy is 10.58%;

When noise magnitude = 0.1, test accuracy is 10.53%;

Based on these implemented models, we continued to build a defense model, using adversarial training. We pick FGSM as the attack model for our deference.

The test accuracy after defense is 13.41%. Higher than the accuracy of the previous model. We believe the slight increase is due to the lack of depth in our model. The model contains only 2 conv blocks, which may be unable to learn too much information.

Here is the loss vs epoch figure:



5. Conclusion

The findings demonstrate the DCNN's capabilities and limitations in handling adversarial attacks on the CIFAR-10 dataset. Based on the testing results, we see that as the noise increases, the accuracy decreases. The reason is, as the value of noise increases, the attack function produces stronger attacks which makes it harder for the model to label them correctly. Therefore, the accuracy of the model

decreases. Despite robust initial performance, the network's vulnerability to sophisticated attacks like PGD underscores the need for continuous enhancements in adversarial defense strategies.

6. Vision Transformer model

(a) Background:

Here we used Vision Transformers (ViTs) to train the CIFAR-10. Unlike CNNs, ViTs do not process the image as a whole but divide it into patches and apply self-attention mechanisms to understand relationships between them. This allows ViTs to capture both local and global contexts efficiently.

(b) Method:

We used PyTorch and Torchvision libraries. The first one is for building and training the neural network models, and the second one is used for loading and transforming the CIFAR-10 dataset. The CIFAR-10 dataset is directly loaded using Torchvision, which also handles the necessary transformations such as resizing, cropping, and normalization to prepare images for the input required by the Vision Transformer.

Other than that, we used a pre-trained model from the ‘timm’ (PyTorch Image Models) library, which could improve the efficiency of our training process, reduce time and improve accuracy.

In the code, the line that specify the loading of ‘timm’ library is:

```
model = timm.create_model('vit_small_patch16_224', pretrained=False, num_classes=10)
```

Where:

Model Selection: 'vit_small_patch16_224' specifies a small Vision Transformer model where the image is expected to be 224x224 pixels, and patches are 16x16 pixels.

Pretraining: The pretrained=False parameter indicates that the model does not load pre-trained weights, making it train from scratch on CIFAR-10.

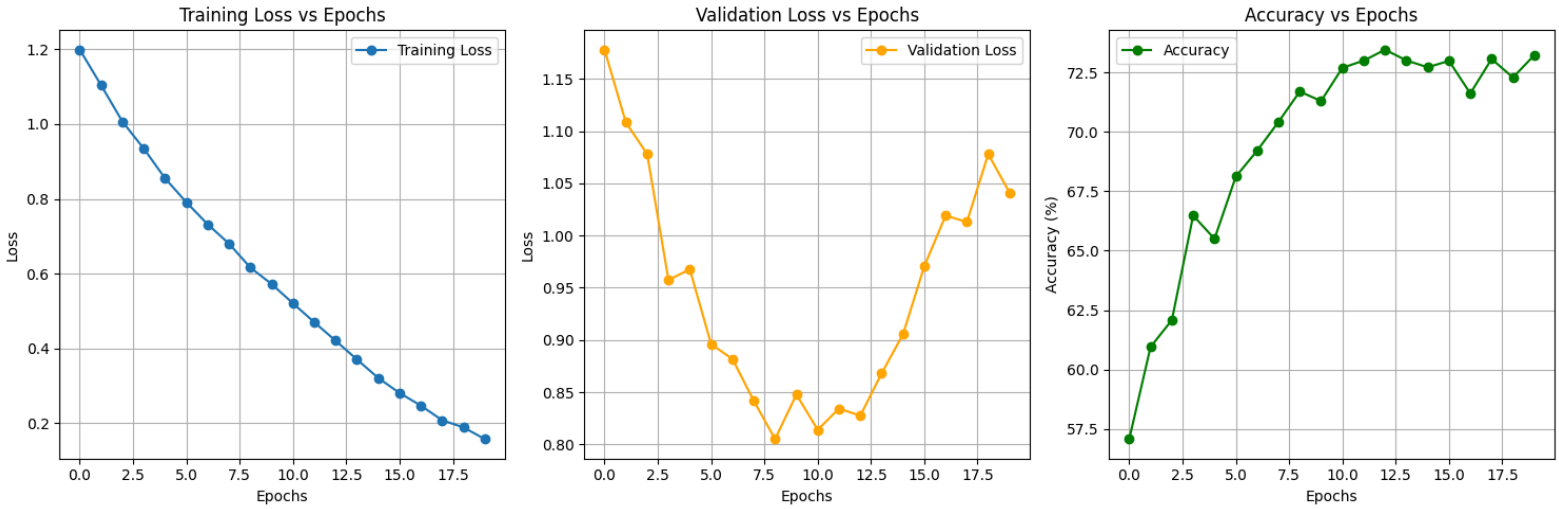
Adaptation to CIFAR-10: Adjusting the final layer to classify 10 classes, as specified by num_classes=10.

(c) Training Process:

Optimizer: Adam with a learning rate of 1e-4.
Loss Function: Cross-Entropy Loss, which is standard for classification tasks.
Epochs: The model is trained for 20 epochs.
Batch Size: Set to 128 to balance the trade-off between memory usage and model performance.

(d) Result:

With a total of 20 epochs, we plotted the result in three plots, which demonstrate the change of Training Loss, Validation Loss, and Accuracy when used to classify testing data.



According to this result, the training loss is decreasing consistently, which indicates the learning process is performing as expected. Also, the accuracy of this model, when tested on testing data, is ascending at the first 12 epochs. However, it started to fluctuate beyond that point.
The highest test accuracy this model can achieve is 73.46%

(e) Conclusion:

The application of Vision Transformers to the CIFAR-10 dataset demonstrates promising results, showcasing the model's ability to leverage its architecture for effective image classification. Throughout the training process, the model exhibited an increase in accuracy, particularly notable in the initial epochs. However, as observed, the accuracy began to fluctuate after 12 epochs. This fluctuation can be attributed to several factors inherent in the training dynamics of deep learning models. One primary reason could be the model approaching its capacity to learn from the given dataset within the constraints of its architecture and hyperparameters. Adjustments such as tuning the learning rate, introducing regularization strategies like dropout or weight

decay, or employing techniques like learning rate scheduling could potentially stabilize the training process and lead to better generalization on the test set.

Extending the number of training epochs may provide the model with more opportunities to refine its weights and biases to better fit the training data. However, care must be taken to monitor for signs of overfitting, likely due to noise in the dataset. Using a more sophisticated validation strategy like k-fold cross-validation, or incorporating early stopping mechanisms could be beneficial.