DEEP LEARNING

CSL 4020

Assignment 9 Report

Transfer Learning

By - PRANAV PANT

B21CS088

Transfer Learning with Regularization Techniques

Introduction:

In the field of computer vision, transfer learning has emerged as a powerful technique to leverage pre-trained models for new tasks, especially when labeled data is scarce.

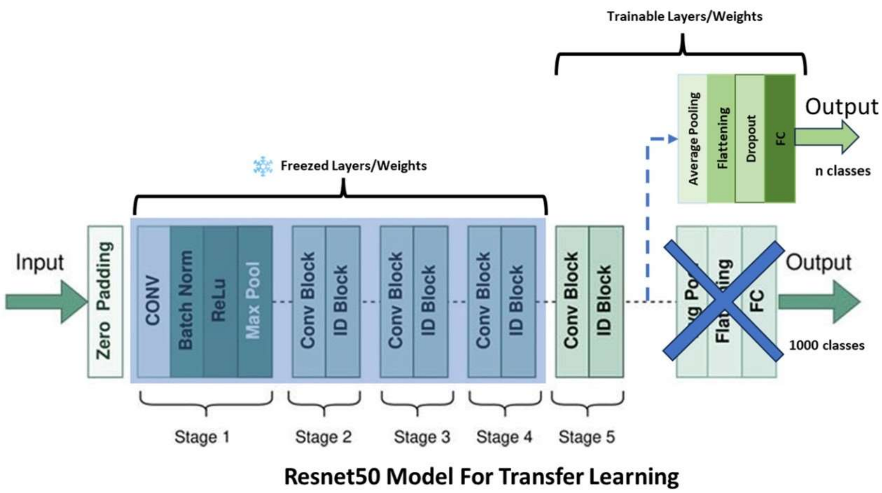

Figure: Resnet for Transfer Learning

This report explores the application of transfer learning using the ResNet-18 architecture on the hymenoptera dataset, which consists of images of ants and bees. Furthermore, the impact of regularization techniques, specifically L1 and L2 regularization, on model performance is investigated. The objective is to enhance model generalization and prevent overfitting by applying these regularization methods.

Dataset details:

The hymenoptera dataset is a subset of a larger dataset containing images of ants and bees. It is structured into training and validation sets, each containing images categorized into two classes:

- **Ants**
- **Bees**

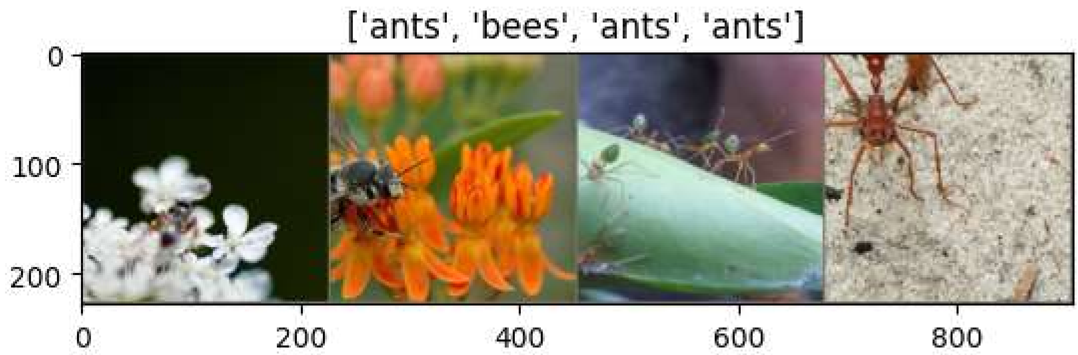

Figure: Plotting some images from the dataset

Data Augmentation and Normalization:

To improve the model's robustness and ability to generalize, data augmentation techniques were applied to the training dataset:

- **RandomResizedCrop**: Randomly crops the input image to a size of 224x224 pixels.
- **RandomHorizontalFlip**: Randomly flips the image horizontally with a probability of 0.5.

Both training and validation datasets were normalized using the mean and standard deviation of the ImageNet dataset:

- **Mean**: [0.485, 0.456, 0.406]
- **Standard Deviation**: [0.229, 0.224, 0.225]

Model Architecture:

ResNet-18 Overview:

ResNet-18 is a convolutional neural network with 18 layers, known for its residual connections that mitigate the vanishing gradient problem. It is pre-trained on the ImageNet dataset, which allows it to extract rich features from images.

Customization for Transfer Learning:

The final fully connected layer of ResNet-18 was modified to match the number of classes in the hymenoptera dataset:

- **Original Output Features**: 1000 (ImageNet classes)
- **Modified Output Features**: 2 (ants and bees)

This was achieved by replacing the original fully connected layer with a new linear layer.

Implementation & Training

Loss Function and Optimizer

- **Loss Function**: Cross-Entropy Loss (nn.CrossEntropyLoss), suitable for multi-class classification tasks.
- **Optimizer**: Stochastic Gradient Descent (optim.SGD) with a learning rate of 0.001 and momentum of 0.9.
- **Learning Rate Scheduler**: Reduces the learning rate by a factor of 0.1 every 7 epochs (lr_scheduler.StepLR).

Regularization Techniques:

1. Without Regularization

The base model was trained without any additional regularization to serve as a benchmark.

2. L1 Regularization

L1 regularization adds the absolute value of the magnitude of coefficients as a penalty term to the loss function.

A *lambda l1* value of 0.03 was used to control the strength of the regularization.

3. L2 Regularization

L2 regularization adds the squared magnitude of coefficients as a penalty term. It was implemented by adding a weight_decay parameter to the optimizer.

BONUS: L1+L2 Regularization

Added both L1 and L2 regularization terms to the model loss for benefit of both the use cases.

Training Procedure:

The model was trained for 25 epochs for each experiment. During each epoch, the following steps were performed:

1. **Forward Pass**: Compute the model's predictions.
2. **Loss Calculation**: Compute the loss using the criterion, adding regularization penalties if applicable.
3. **Backward Pass**: Compute gradients with respect to the loss.
4. **Optimizer Step**: Update model weights.
5. **Learning Rate Adjustment**: Update the learning rate according to the scheduler.

Performance metrics, including training and validation loss and accuracy, were recorded after each epoch.

Results:

Performance Metrics:

The table with overall precision, recall, F1-score, and accuracy:

Regularization Type	Overall Precision	Overall Recall	Overall F1-Score	Accuracy (%)
Without Regularization	0.95	0.95	0.95	94.77
With L1 Regularization	0.86	0.83	0.83	83.66
With L2 Regularization	0.90	0.90	0.90	90.20
L1+L2 Regularization	0.93	0.93	0.93	92.81

Loss and Accuracy Curves:

- **Without Regularization**

Figure: Plotting Loss / epoch and Accuracy / epoch graphs.

- **With L1 Regularization**

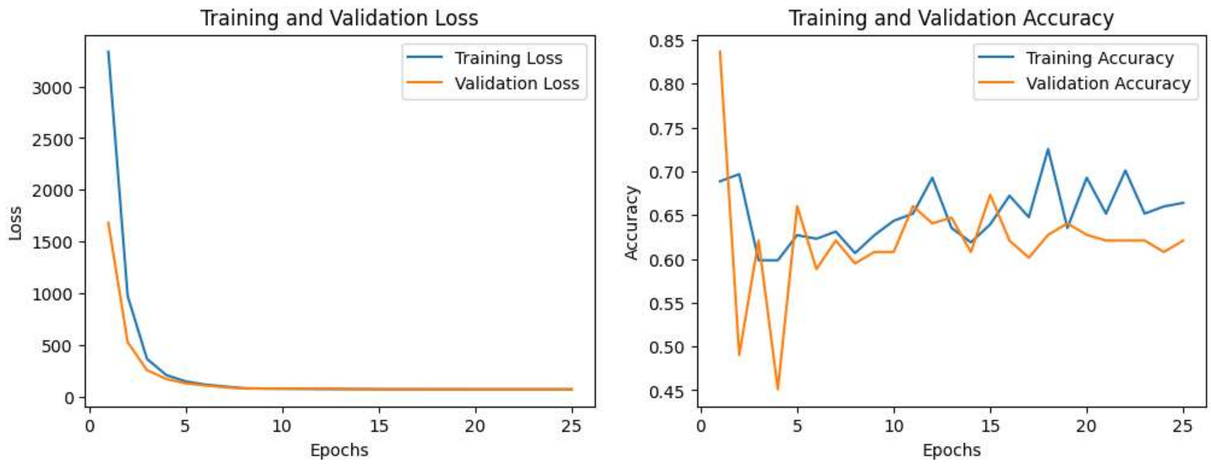

Figure: Plotting Loss / epoch and Accuracy / epoch graphs.

- **With L2 Regularization**

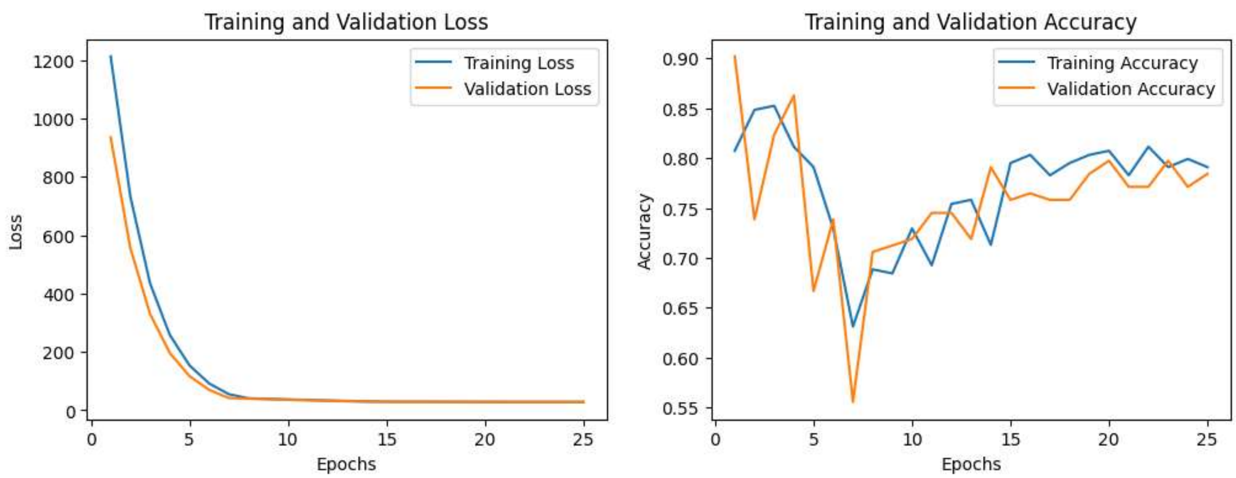

Figure: Plotting Loss / epoch and Accuracy / epoch graphs.

BONUS: **With Both L1, L2 regularization (ElasticNet):**

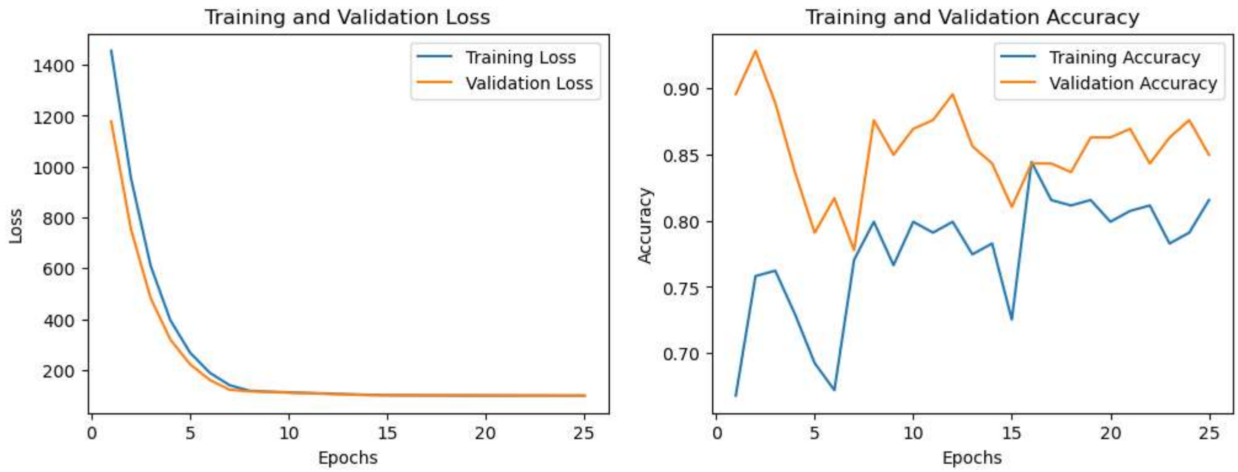

Figure: Plotting Loss / epoch and Accuracy / epoch graphs.

Visualizing Predictions:

Figure: Predicting some test images from model with l1 regularization.

Figure: Predicting some test images from model with l1+l2 regularization.

Conclusion:

The experiments demonstrate the impact of regularization techniques on the performance of a transfer learning model:

- **Without Regularization**: The model achieved the highest validation accuracy of 94.77%, indicating good generalization on the validation set.
- **With L1 Regularization**: The validation accuracy decreased to 83.66%. The L1 penalty might have been too strong (lambda_l1 = 0.03), leading to underfitting.
- **With L2 Regularization**: The model achieved a validation accuracy of 90.20%. While there was a reduction compared to the model without regularization, L2 regularization helped in preventing overfitting to some extent.
- **With L1+L2 Regularization:** This model combining both the L1 and L2 regularization techniques also performs very good with a validation accuracy of **92.81%** and prevents overfitting.

Overall, regularization is a crucial technique for enhancing model generalization, but it requires careful tuning to achieve optimal results.

---*Report Ends*--