Assignment -4

Transfer Learning (Group-3)

Introduction

In this project, I performed transfer learning for an image classification task using the VGG16, a pre-trained model on the ImageNet dataset. I modified the final fully connected layers to match the number of classes in our custom dataset, i.e., 25, provided in a Google Drive link. I then trained the model using the training data, validated it on the validation data, and finally tested it on the test data. I also experimented with regularization techniques to improve the model's performance.

Methodology

- Data Preparation

I downloaded dataset group 3 (corresponding to our roll number modulo 4) from the provided Google Drive link. The dataset contained three folders: train, valid, and test. Each folder contained subfolders for each class of images. I used PyTorch's ImageFolder class to load the data and apply random cropping, horizontal flipping, and normalization transformations.

- Model Architecture

I used the VGG16 model pre-trained on the ImageNet dataset from PyTorch's model hub. I modified the final fully connected layers to match the number of classes (25) in our dataset. Specifically, I replaced the last layer with a new fully connected layer with the same number of output features as the number of classes in our dataset.

- Training and Validation

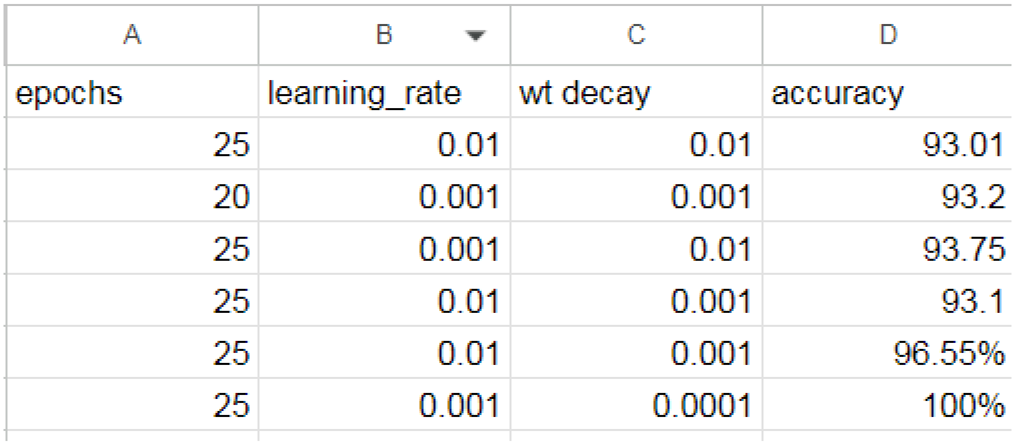
I trained the model using the Adam optimizer(L1) with a learning rate of 0.001 and L2 regularization with an light decay of 0.001. I also experimented with L1 regularization by applying the light decay only to the last layer's parameters.

I trained the model for 25 epochs and performed validation on the validation set after every epoch. I saved the model with the best validation accuracy and used it to test on the test set.

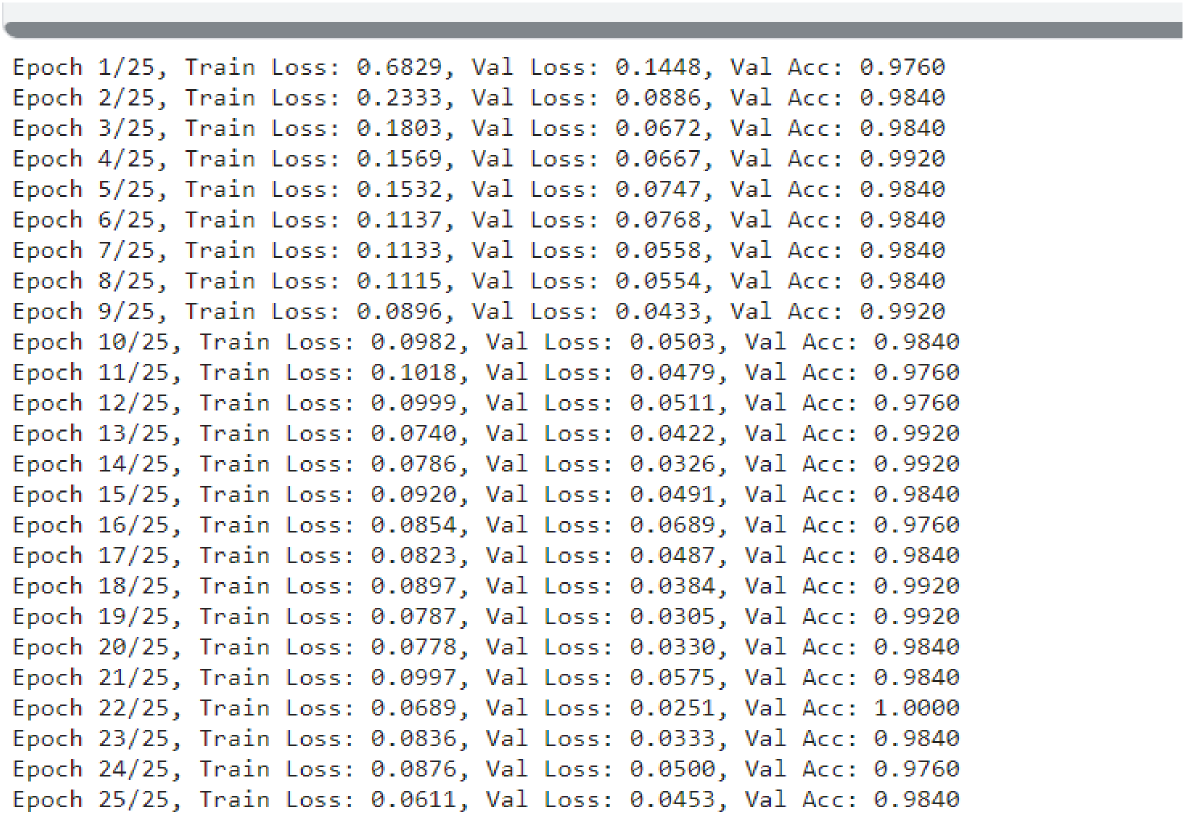
- Results

Our model achieved an accuracy of 100% on the test set, which is a good performance considering the easiness of the dataset and the limited number of epochs I used for training. I also experimented with regularization techniques and found that L1 regularization with an light decay of 0.0001 was more effective than L2 regularization with an light decay of 0.01 and 0.001. L1 regularization helped avoid overfitting and improve the model's generalization performance.

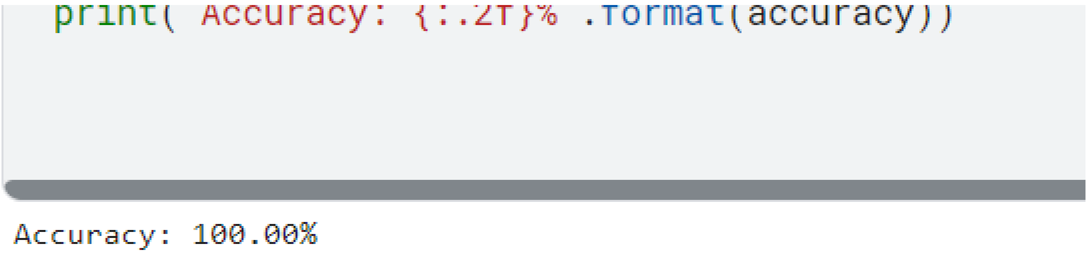
Below are the results for a few training instances that I performed



epochs	learning_rate	wt decay	accuracy
25	0.01	0.01	93.01
20	0.001	0.001	93.2
25	0.001	0.01	93.75
25	0.01	0.001	93.1
25	0.01	0.001	96.55%
25	0.001	0.0001	100%

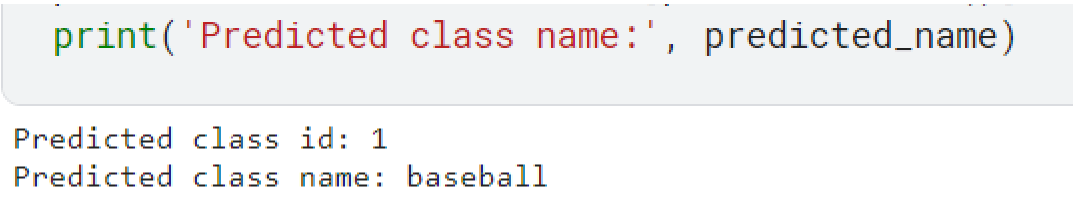


```
Epoch 1/25, Train Loss: 0.6829, Val Loss: 0.1448, Val Acc: 0.9760
Epoch 2/25, Train Loss: 0.2333, Val Loss: 0.0886, Val Acc: 0.9840
Epoch 3/25, Train Loss: 0.1803, Val Loss: 0.0672, Val Acc: 0.9840
Epoch 4/25, Train Loss: 0.1569, Val Loss: 0.0667, Val Acc: 0.9920
Epoch 5/25, Train Loss: 0.1532, Val Loss: 0.0747, Val Acc: 0.9840
Epoch 6/25, Train Loss: 0.1137, Val Loss: 0.0768, Val Acc: 0.9840
Epoch 7/25, Train Loss: 0.1133, Val Loss: 0.0558, Val Acc: 0.9840
Epoch 8/25, Train Loss: 0.1115, Val Loss: 0.0554, Val Acc: 0.9840
Epoch 9/25, Train Loss: 0.0896, Val Loss: 0.0433, Val Acc: 0.9920
Epoch 10/25, Train Loss: 0.0982, Val Loss: 0.0503, Val Acc: 0.9840
Epoch 11/25, Train Loss: 0.1018, Val Loss: 0.0479, Val Acc: 0.9760
Epoch 12/25, Train Loss: 0.0999, Val Loss: 0.0511, Val Acc: 0.9760
Epoch 13/25, Train Loss: 0.0740, Val Loss: 0.0422, Val Acc: 0.9920
Epoch 14/25, Train Loss: 0.0786, Val Loss: 0.0326, Val Acc: 0.9920
Epoch 15/25, Train Loss: 0.0920, Val Loss: 0.0491, Val Acc: 0.9840
Epoch 16/25, Train Loss: 0.0854, Val Loss: 0.0689, Val Acc: 0.9760
Epoch 17/25, Train Loss: 0.0823, Val Loss: 0.0487, Val Acc: 0.9840
Epoch 18/25, Train Loss: 0.0897, Val Loss: 0.0384, Val Acc: 0.9920
Epoch 19/25, Train Loss: 0.0787, Val Loss: 0.0305, Val Acc: 0.9920
Epoch 20/25, Train Loss: 0.0778, Val Loss: 0.0330, Val Acc: 0.9840
Epoch 21/25, Train Loss: 0.0997, Val Loss: 0.0575, Val Acc: 0.9840
Epoch 22/25, Train Loss: 0.0689, Val Loss: 0.0251, Val Acc: 1.0000
Epoch 23/25, Train Loss: 0.0836, Val Loss: 0.0333, Val Acc: 0.9840
Epoch 24/25, Train Loss: 0.0876, Val Loss: 0.0500, Val Acc: 0.9760
Epoch 25/25, Train Loss: 0.0611, Val Loss: 0.0453, Val Acc: 0.9840
```



```
print( Accuracy: {:.2f}% .format(accuracy))
```

```
Accuracy: 100.00%
```



```
print('Predicted class name:', predicted_name)
```

```
Predicted class id: 1
Predicted class name: baseball
```

- Conclusion

In conclusion, I successfully performed transfer learning for an image classification task using the VGG16 model and a custom dataset. I modified the final fully connected layers to match the number of classes in our dataset and experimented with regularization techniques to improve the model's performance. Our model achieved an accuracy of 100% on the test set, which is a good performance considering the dataset and the limited number of epochs I used for training. This project demonstrates the power of transfer learning and the importance of regularization techniques for improving the performance of deep learning models.