Exercise 5: Transfer Learning with ResNet

Objective

Leverage transfer learning to train a pre-trained ResNet model on one of the datasets used in Exercise 4. Evaluate and compare the performance metrics of the transfer learning approach with your CNN from Exercise 4. Document your findings and conclusions.

Instructions

1. Setup and Data Preparation

1. Choose a dataset from the options provided in Exercise 4:

- Fashion-MNIST
- Street View House Numbers (SVHN)
- Intel Image Classification
- Tiny ImageNet

2. Preprocess the dataset:

- Resize images to match the input dimensions required by ResNet (e.g., 224 x 224).

- Normalize the dataset using the mean and standard deviation of ImageNet:

```
transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
```

3. Apply data augmentation:

- Add transformations such as random horizontal flips, rotations, and color jitter to improve generalization.

2. Model Building

1. Use a pre-trained ResNet model (e.g., ResNet18 or ResNet50) from torchvision.models:

```
from torchvision.models import resnet18
model = resnet18(pretrained=True)
```

2. Modify the model:

Replace the fully connected layer to match the number of classes in your dataset:

```
model.fc = nn.Linear(model.fc.in_features, num_classes)
```

3. Freeze all convolutional layers for initial training:

```
for param in model.parameters():
    param.requires_grad = False
```

4. Enable fine-tuning after a few epochs by unfreezing the earlier layers.

3. Training

1. Define hyperparameters:
- Learning rate, batch size, number of epochs, optimizer, and scheduler.
- Use CrossEntropyLoss for classification.

2. Train the model:
- Start with the fully connected layer and fine-tune the convolutional layers after observing initial results.

3. Monitor metrics:
- Track accuracy, precision, recall, and F1-score during training.

4. Save the best model using torch.save().

4. Evaluation

1. Test the model:
- Evaluate the model on the test set.
- Compute metrics such as accuracy, precision, recall, and F1-score.

2. Visualize performance:
- Generate a confusion matrix.
- Plot training and validation loss/accuracy curves.

5. Comparison and Conclusions

1. Compare transfer learning performance with your custom CNN from Exercise 4:
- Highlight improvements in training time, convergence, and accuracy.
- Discuss scenarios where transfer learning excels versus training from scratch.

2. Document key findings:
- Write a summary explaining how the pre-trained ResNet impacted the results.
- Include any challenges encountered during training or fine-tuning.

Deliverables

1. Jupyter Notebook:
- Well-documented code with outputs showing training progress and evaluation results.

2. Report Document:
- Summarize the methodology, results, and key comparisons between the CNN and ResNet.
- Include:
  - Plots of training and validation metrics.
  - Confusion matrix.
  - Table comparing CNN and ResNet metrics.
  - Your conclusions on transfer learning effectiveness.