Image Classification with Convolutional Neural Networks

Part 1: Training and Evaluating Your Model

- Parametric details: The models have convolutional layers with specific input and output channels, stride, and padding. We use ReLU activation function, CrossEntropyLoss as the loss function, and Adam optimizer.
- Residual connections: Residual connections are implemented by adding the input to the output of each convolutional layer in the model.
- Accuracy calculation: Accuracy is calculated by comparing predicted labels with true labels and calculating the percentage of correct predictions.
- Graph of loss and accuracy change: We can plot the loss and accuracy change over epochs for different learning rates and batch sizes using the **plot_metrics** function.
- Integrating dropout: Dropout is added after the fully connected layer in both models to improve generalization. We can explore different dropout rates and evaluate the new accuracies on the validation and test sets.
- Confusion matrix: We can plot a confusion matrix for the best model's predictions using the **confusion_mat** function.

Part 2: Transfer Learning with CNNs

- Fine-tuning: Fine-tuning involves adapting a pre-trained network to a new task by freezing most layers and updating only the last few layers. We freeze the rest because lower layers learn general features. We can freeze layers in the ResNet-18 model by setting **requires_grad** to False for their parameters.
- Training cases: We explore two cases: training only the fully connected (FC) layer and freezing the rest, and training the last two convolutional layers and the FC layer while freezing the rest. We evaluate accuracy on the validation and test sets for both cases.

Cavit Bora Öztekeşin 2200356855