Homework 4

Comparative Analysis of Vision Transformer and SWIN Models for Image Classification

Due: 5/25/2025

Objective:

To implement and compare the performance of Vision Transformer and SWIN models on the CIFAR-10 dataset for object classification, and to analyze their decision-making processes using Grad-CAM visualization.

Instructions

1. Data Preparation
   - Load the CIFAR-10 dataset using a deep learning framework such as PyTorch or TensorFlow.
   - Preprocess the images:
   - Normalize pixel values (e.g., using mean and standard deviation of the dataset).
   - Resize images from 32x32 to 224x224 to match the input size expected by pre-trained models.
   - Utilize the standard training and testing splits provided by the CIFAR-10 dataset.

2. Model Selection and Fine-tuning
   - Select pre-trained Vision Transformer (ViT) and SWIN Transformer models from a library like `timm` (PyTorch Image Models).
   - Modify the classification head of each model to output 10 classes (corresponding to CIFAR-10 categories) instead of the default 1000 (ImageNet classes).
   - Fine-tune the models on the CIFAR-10 training set:
     - Option 1: Fine-tune the entire model if computational resources permit.
     - Option 2: Freeze earlier layers and fine-tune only the classification head and later layers to reduce computational demand.

3. Training
   - Train both models using suitable optimizers (e.g., AdamW) and learning rate schedules (e.g., cosine annealing).
   - Monitor training progress by tracking loss and accuracy on both training and validation sets.
   - Save the model checkpoints that achieve the highest validation accuracy.

4. Evaluation
   - Evaluate the fine-tuned models on the CIFAR-10 testing set.
   - Compute and report classification accuracy and other relevant metrics (e.g., top-1 error rate, confusion matrix).

5. Grad-CAM Visualization
   - Randomly select one image from the CIFAR-10 testing set.
   - Apply Grad-CAM to both models to visualize the regions of the selected image that influence their classification decisions.
   - Use a library such as `pytorch-gradcam` to implement Grad-CAM:
     - https://github.com/jacobgil/pytorch-grad-cam.git
     - For Vision Transformer, target the gradients of the class token or the last attention layer.
     - For SWIN, target the feature maps from the last hierarchical stage.
   - Generate and display Grad-CAM heatmaps for both models on the chosen image.

6. Analysis and Comparison
   - Compare the classification performance of Vision Transformer and SWIN based on accuracy and computational efficiency.
   - Analyze the Grad-CAM visualizations to identify differences in how each model attends to various parts of the image.
   - Discuss the implications of these attention patterns on the models' decision-making processes.
   - Reflect on the strengths and limitations of each model for the CIFAR-10 classification task.

Deliverables:

- Report: A detailed document including:
  - Methodology (data preparation, model setup, training process).
  - Experimental results (accuracy, metrics, visualizations).
  - Analysis and comparison of the two models.
- Code Implementation: Complete scripts for training, evaluation, and visualization, including comments for clarity. Push them to your GitHub.
- Grad-CAM Visualizations: Heatmaps generated for the randomly selected test image from both models.

Notes:

- Optional Exploration: Experiment with different hyperparameters (e.g., learning rate, batch size) or model variants if time and resources allow.

Example code: see attached Jupyter notebook