AP Lab03 Tensorflow and PyTorch

Lab Instructions - Prof. Tobias Schaffer

Objective

In this lab, you will:

- Build and train the same small neural network model using both TensorFlow and PyTorch.
- Measure and compare training time and performance.
- Convert the trained models into lightweight formats suitable for embedded deployment: TensorFlow Lite and ONNX.

Task 1: Model Implementation and Training

Dataset

Use the MNIST dataset of handwritten digits (28x28 grayscale images, 10 classes).

Architecture

- Flatten input (784 features)
- Dense layer with 64 ReLU units
- Output layer with 10 units (softmax for TF, logits for PyTorch)

Instructions

1. Load and normalize MNIST data.
2. Implement the model in TensorFlow using `tf.keras.Sequential`.
3. Implement the model in PyTorch using a custom `nn.Module`.
4. Train both models for 5 epochs.
5. Measure training time.

Task 2: Inference and Evaluation

- Run inference on the test set using both models.
- Report test accuracy and inference time.
- Use TensorFlow's `model.evaluate()` and PyTorch's `model.eval()` + `torch.no_grad()`.

Task 3: Model Conversion

TensorFlow

1. Convert the trained model to TensorFlow Lite using `TFLiteConverter`.
2. Save the converted model as `model.tflite`.

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)
```

PyTorch

1. Export the model to ONNX format.
2. Use dummy input with correct shape (e.g. `torch.randn(1, 784)`).
3. Save as `model.onnx`.

```
dummy_input = torch.randn(1, 784)
torch.onnx.export(model, dummy_input, "model.onnx",
                  input_names=["input"], output_names=["output"])
```

Submission

Please submit the following:

- Python scripts for both implementations.
- Training and inference logs.
- Exported model files: `model.tflite` and `model.onnx`.
- Short report comparing the frameworks in terms of:
  - Code structure and development experience
  - Training and inference speed
  - Ease of model export

A sample implementation (with missing parts to be filled) can be found in the file:
`Lab03 TensorFlow vs PyTorch.ipynb`