

Deep Learning Assignment 3

In this assignment, you are required to train and evaluate multiple pre-trained models from `torchvision.models` on vision datasets such as MNIST, FashionMNIST, CIFAR-10, or CIFAR-100. You must report your code along with a table summarizing the accuracy of each model on the chosen dataset.

Assignment Steps:

1. Dataset Selection: Choose one of the following datasets:

- MNIST
- FashionMNIST
- CIFAR-10
- CIFAR-100
- SVHN (Street View House Numbers)

Use the appropriate transformations to prepare the dataset (e.g., resizing to 224x224 and normalization).

2. Pre-trained Models: Use pre-trained models from `torchvision.models`. You must train and evaluate at least three of the following models:

- ResNet18
- AlexNet
- VGG16
- MobileNetV2
- DenseNet121
- EfficientNet-B0
- GoogLeNet

Ensure the final layer is modified to match the number of classes in the chosen dataset.

3. Training and Evaluation:

- Write PyTorch code to train each model for at least 5 epochs.
- Evaluate each model on the test dataset and calculate its accuracy.
- Ensure your code runs efficiently by utilizing GPU when available (on Kaggle, Colab, etc..).

Use the following training components:

- Optimizer: Adam
- Loss Function: CrossEntropyLoss
- Batch Size: 32
- Learning Rate: 0.001

4. Report Results:

Summarize the test accuracy of all trained models in a table as follows:

Model	Test Accuracy (%)
-------	-------------------

ResNet18	XX.X
VGG16	XX.X
MobileNetV2	XX.X

Include your PyTorch code and the table in your final report.

Submission Instructions:

1. Submit your report as a single PDF or Word file.
2. Attach your Jupyter Notebook (`.ipynb`) with the submission.
3. Ensure all results are reproducible, and your code is well-documented.
4. Mention your dataset and any modifications you applied to the pre-trained models.

Hints

1. Use the `torchvision.transforms` module to resize and normalize images
2. Use the `torch.nn.CrossEntropyLoss()` loss function and `torch.optim.SGD` or `torch.optim.Adam` optimizers.
3. Use `torch.cuda.is_available()` to check if GPU support is available.
4. Example code snippets for modifying the model's last layer:

```
import torchvision.models as models
```

```
import torch.nn as nn
```

```
model = models.resnet18(pretrained=True)
```

```
num_fts = model.fc.in_features
```

```
model.fc = nn.Linear(num_fts, num_classes) # Replace last layer
```

5. You can use libraries like `tqdm` to display progress bars for training loops.
6. Please follow the tutorial prepared for this assignment:
<https://colab.research.google.com/drive/1BmYsCF9WANNCDrktNWE6guRxvZ0Obcyr?usp=sharing>