Deep Learning Report

Lab Assignment - 6

Ayush Abrol
B20AI052

Question 01

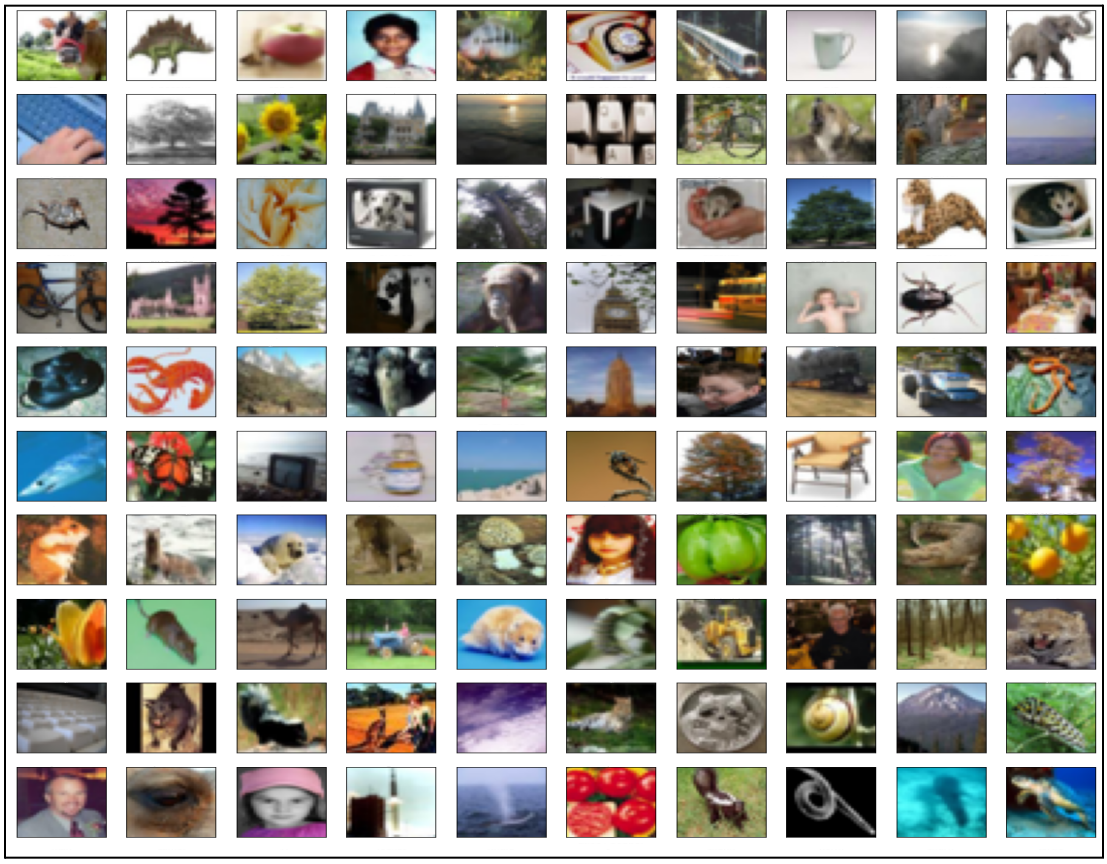
Aim: Based on the lecture by Dr. Anush on DLOPs, you have to perform the following experiments :

- Load and preprocessing CIFAR100 dataset using standard augmentation and normalization techniques
- Train the following models for 50 epoch and at the same time profile the model using Tensorboard during the training step
    - ResNet-34
    - DenseNet-121
    - EfficientNet-B0
    - ConvNeXt-T
- Then perform the following model inferencing techniques on the above listed models.
    - ONNX ; ONNX Quantized
    - Torchscript
- Report the model size and average execution time before and after performing the above mentioned inferencing techniques on the test dataset.

Analyze the models based on their architecture and inferencing techniques.

Procedure:

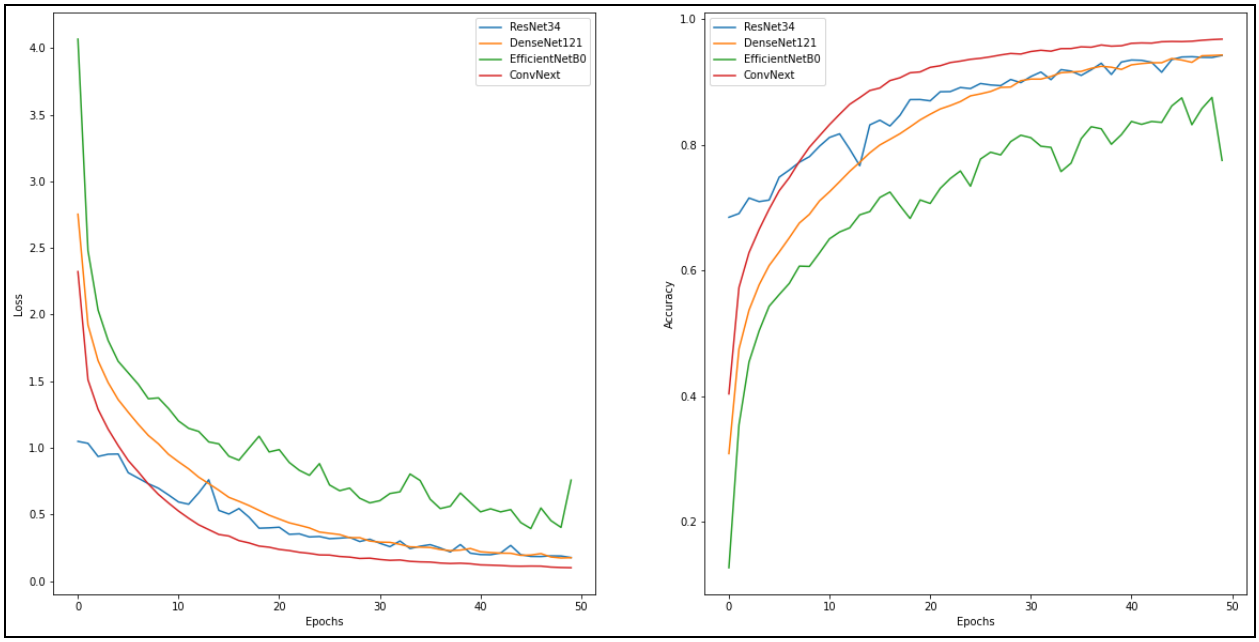
- The CIFAR100 dataset, which consists of 60,000 32x32 color images categorized into 100 classes, is what we'll be using for this lab. Twenty super-classes made up of five fine-grained classes each are created from the classes. The dataset was created. For designing and testing machine learning methods as well as for image recognition tasks. Compared to CIFAR-10, which only has 10 classes, the visuals in CIFAR-100 are more intricate and varied. A common benchmark for assessing the effectiveness of image classification methods is CIFAR-100.
- The dataset is normalized, rotated, and horizontally flipped using typical augmentation techniques.



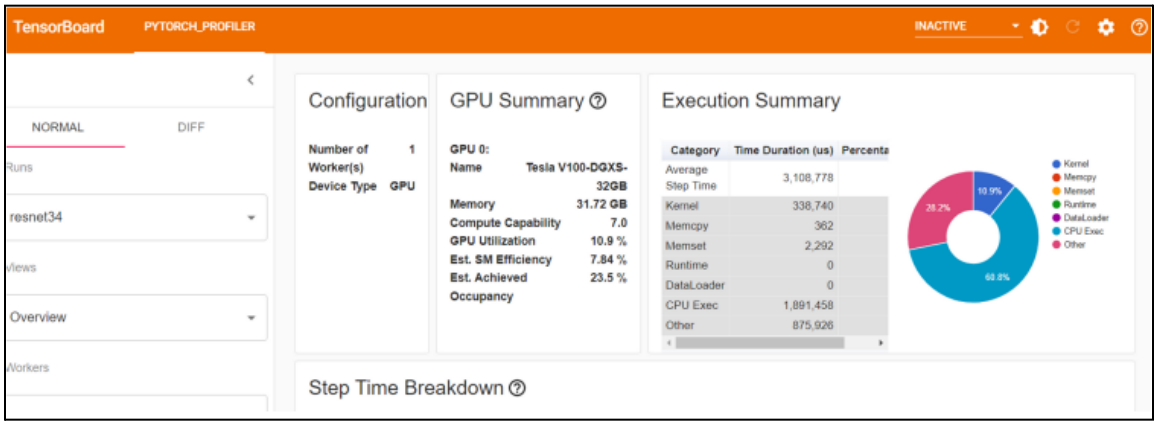
- For the CIFAR100 dataset, we train 4 models from scratch without using any pre-trained weights. Resnet-34, Densenet-121, EfficientNet-B0, and ConvNext-Tiny are the models in question. A learning rate of 1e-3 and the Adam optimizer are used during the 50-epoch training process.

Model	Train Accuracy	Train Loss
Resnet-34	94.20	0.1770
DenseNet-121	94.26	0.1757
EfficientNet-B0	77.51	0.7580
ConvNext-Tiny	96.79	0.1007

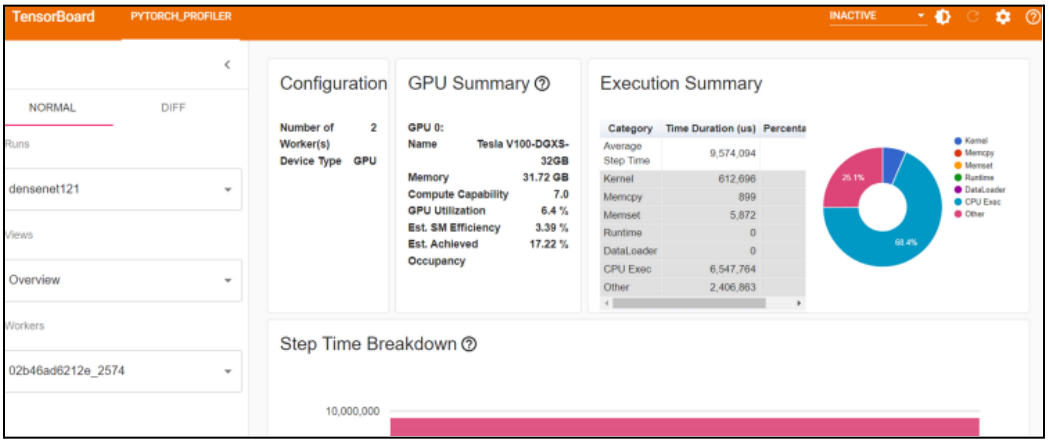
- Our best model for this localized training session is ConvNext-Tiny

- EfficientNet-B0 is challenging to directly train, as the loss appears to not converge at all, even after 50 epochs. The accuracy also reveals a low training capacity. This might be a problem because of the local dataset configuration, and training could be improved by using larger images and approaches like learning rate scheduling and warmups, among others.
- 

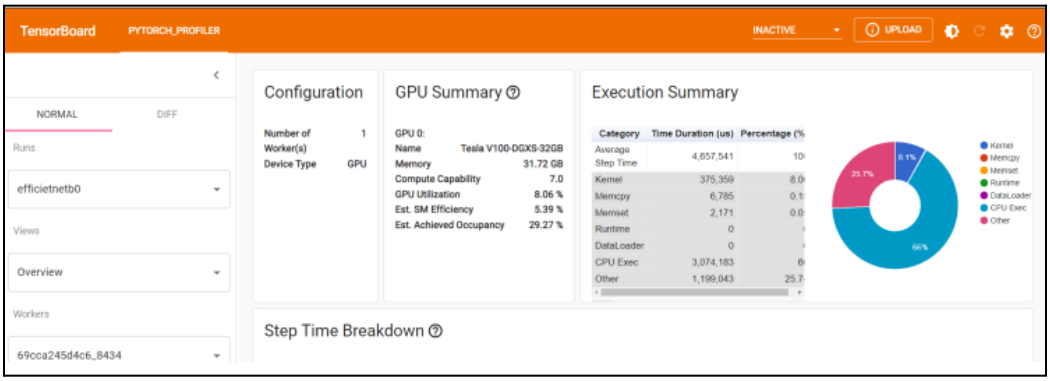
- Tensorboard Profiling
    - Resnet-34



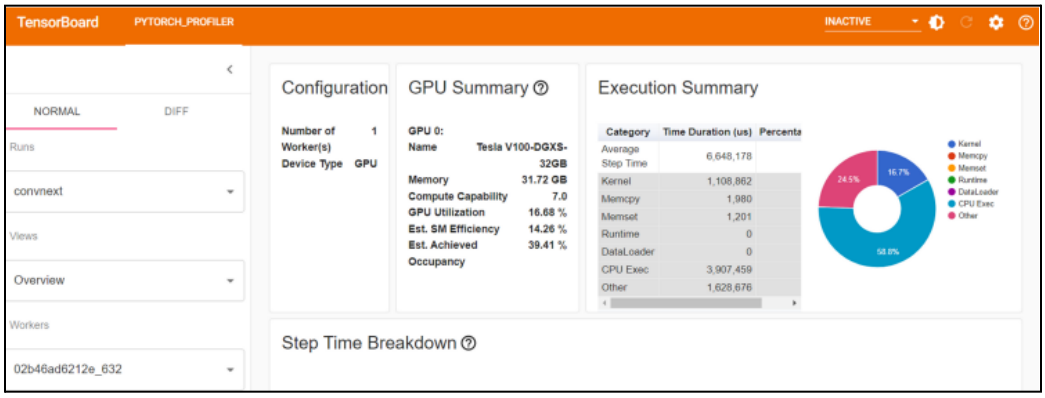
- DenseNet-121



- EfficientNet-B0



- ConvNext-T



- Inferencing using TorchScript and ONNX

  Resnet-34

```
Pytorch model size (MB): 81.52365970611572
TorchScript model size (MB): 81.5255479812622
Optimzed TorchScript model size (MB): 81.5255479812622
```

```
Average runtime of Pytorch resnet34 Model in  current using device: 24.678371936750388
Average runtime of TorchScript resnet34 Model in current using device: 30.25847188607109
Average runtime of optimzed TorchScript resnet34 Model in current using device: 24.445588114114017
```

```
Pytorch resnet34 model size (MB): 81.52365970611572
ONNX full precision resnet34 model size (MB): 81.4544776781152
ONNX quantized resnet34 model size (MB): 20.880047779815674
```

```
Average runtime of ONNX resnet34 Model in current using device: 9.257106645512792
Average runtime of ONNX Optimized resnet34 Model in current using device: 9.996364405139238
Average runtime of ONNX Quantized resnet34 Model in current using device: 40.58078600000148
```

  DenseNet-121

```
Pytorch densenet121 model size (MB): 27.496817588806152
TorchScript densenet121 model size (MB): 27.503074645996094
Optimzed TorchScript densenet121 model size (MB): 0.00049304962158203125
```

```
Average runtime of Pytorch densenet121 Model in  current using device: 23.843062670908417
Average runtime of TorchScript densenet121 Model in current using device: 25.905542075818204
Average runtime of optimzed TorchScript densenet121 Model in current using device: 28.42410749342636
```

```
Pytorch densenet121 model size (MB): 31.015860557556152
ONNX full precision densenet121 model size (MB): 30.811427116394043
ONNX quantized densenet121 model size (MB): 8.444693565368652
```

```
Average runtime of ONNX densenet121 Model in current using device: 8.817258075891282
Average runtime of ONNX densenet121 Optimized Model in current using device: 10.609376873893595
Average runtime of ONNX Quantized densenet121 Model in current using device: 24.485979923651307
```

Efficient-B0

```
Pytorch efficientnet_b0 model size (MB): 20.460755348205566
TorchScript efficientnet_b0 model size (MB): 20.464106559753418
Optimzed TorchScript efficientnet_b0 model size (MB): 20.46341609954834
```

```
Average runtime of Pytorch efficientnet_b0 Model in  current using device: 25.362668860815333
Average runtime of TorchScript efficientnet_b0 Model in current using device: 25.839583923986044
Average runtime of optimzed TorchScript efficientnet_b0 Model in current using device: 40.81272425321026
```

```
Pytorch efficientnet_b0 model size (MB): 20.460755348205566
ONNX full precision efficientnet_b0 model size (MB): 20.164748191833496
ONNX quantized efficientnet_b0 model size (MB): 5.3798065185546875
```

```
Average runtime of ONNX efficientnet_b0 Model in current using device: 7.188476936582585
Average runtime of ONNX Optimized efficientnet_b0 Model in current using device: 4.18058241774804
Average runtime of ONNX Quantized efficientnet_b0 Model in current using device: 29.086621582365883
```

ConvNext-T

```
Pytorch convnextt model size (MB): 109.12201404571533
TorchScript convnextt model size (MB): 109.1235933303833
Optimzed TorchScript convnextt model size (MB): 109.123242378234 86
```

```
Average runtime of Pytorch convnextt Model in  current using device: 26.032809683311836
Average runtime of TorchScript convnextt Model in current using device: 24.46034215208323
Average runtime of optimzed TorchScript convnextt Model in current using device: 27.721071557008848
```

```
Pytorch convnextt model size (MB): 109.12201404571533
ONNX full precision convnextt model size (MB): 109.16754913330078
ONNX quantized convnextt model size (MB): 27.682329177856445
```

```
Average runtime of ONNX convnextt Model in current using device: 24.26227137964562
Average runtime of ONNX Optimized convnextt Model in current using device: 32.199092670948
Average runtime of ONNX Quantized convnextt Model in current using device: 35.507991126848275
```

- Analyzing the models based on their architecture and inferencing techniques

We discovered through the aforementioned studies that Resnet-34 infers conclusions more quickly than Densenet-121. This is consistent with both models' topologies, as Resnet34 has 34 layers whereas Densenet-121 has 121 layers and performs more computations per forward pass. Furthermore, we discover that EfficientNet-inference B0's timings are equivalent to those of Densenet-121 because it is a result of compound scaling in its design, which scales its depth, width, and resolution in order to take into consideration the trade-off of only using deeper models. This characteristic of its architecture makes it slower than Resnet-34 to infer conclusions, but comparable to Densenet-121 and ConvNext-T. Again, this is a result of similar architecture and parameter sizes. Resnet blocks are similar to ConvNext blocks, but ConvNext blocks have higher scaled parameters and convolutional resolution. Hence, the longer inference times.

—----------END—----------