Deep Learning Report

Lab Assignment - 7

Ayush Abrol
B20AI052

Question 01

Aim:

In this assignment we were required to train a Convolutional Neural Network on the following using slurm and submit the jobs.

a. If the last digit of your roll number is even:

Train **ResNet-18 on FashionMNIST dataset.**

- Download the datasets and extract them to folders on the GPU server and preprocess the data. (Use default PyTorch dataloader function mentioning the dataset path and utilize different transforms for preprocessing)
- Set the loss function, optimizer, and metrics and compile the model
- Use Slurm to submit a job for training the model on the GPU server
- Also measure the training time (use timeit module for instance) for 10 and 15 epochs
- Try to identify the set of hyperparameters that results in similar performance as compared to the best performance in the previous step but with lower training time.

Procedure:

- Imported the following libraries:
    - numpy
    - pandas
    - matplotlib.pyplot
    - torch (torch, torch.nn, torch.nn.functional, torch.optim)
    - torchvision (datasets, transforms, models)
    - copy
    - timeit
    - warnings
- Created transform function to transform the data into tensors and normalize the data and added Resize, CenterCrop, RandomHorizontalFlip, RandomRotation to the transform function.

- Downloaded the FashionMNIST data from the torchvision.datasets and applied the transform function to the data.
- Set the environment to cuda:0 GPU so that later in the hpc server GPU could be allocated for the code to run.
- Created train and test data loaders.

```
Train data shape:  torch.Size([60000, 28, 28])
Test data shape:  torch.Size([10000, 28, 28])
Train labels shape:  torch.Size([60000])
Test labels shape:  torch.Size([10000])
```

- Displayed the images from the data loader.
- Imported the pretrained ResNet-18 model from torchvision.models and fine tuned the model to classify the images into 10 classes.
- Defined the loss function as CrossEntropyLoss and optimizer as Adam.
- Defined the train and test functions.
- Trained the model for 10 epochs and trained another model for 15 epochs.
- Plotted the training and validation loss and accuracy for both the models.
- Calculated the time taken for training the model for 10 epochs and 15 epochs and their test accuracies as well.
- Times:
    - For 10 epochs: 690.702s
    - For 15 epochs: 1133.594s
- Test Accuracies:
    - For 10 epochs: 93.00%
    - For 15 epochs: 93.64%
- Saved both the models using torch.save().
- Performed hyperparameter tuning on the model trained for 10 epochs and found the best learning rate, best momentum and best weight decay using SGD optimizer and saved the results in a csv file.
- Displayed the model with the best learning rate, best momentum and best weight decay.

- When the code was done, I converted the code to a .py file and and created a batch file which is as follows:

```
ayushabrol@pop-os:~/Desktop/Lab 7 Slurm$ cat B20AI052_Lab_Assignment_7.sh
#!/bin/bash
#SBATCH --job-name=test_job # Job name
#SBATCH --partition=gpu2 #Partition Name
#SBATCH --nodes=1 # Run all processes on a single node
#SBATCH --ntasks=1 # Run a single task
#SBATCH --cpus-per-task=1# Number of CPU cores per task
#SBATCH --gres=gpu
#

sleep 3
echo "Executing the job by B20AI052"
module load python/3.8

sleep 3
nvidia-smi

sleep 3
python3 B20AI052_Lab_Assignment_7.py
```

- Logged in to the hpc server using ssh b20ai052@172.25.0.15 after authenticating using my password for the same.
- Used the following command to copy local files(.py and .sh) to the server directory:
    - scp -r /home/ayushabrol/Desktop/Lab\ 7\ Slurm/ b20ai052@172.25.0.15:../b20ai052/Lab_7
- Checked the server info using sinfo:

```
[b20ai052@hpclogin Lab 7 Slurm]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
gpu          up 20-00:00:0      1    mix gpu1
gpu2         up 20-00:00:0      1    mix gpu2
test         up 2-00:00:00      1  alloc cn20
small*       up 5-00:00:00      3  alloc cn[01-03]
```

- Looked at the current queue using: watch squeue command:

```
[b20ai052@hpclogin Lab 7 Slurm]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             40004       gpu       50 patanvad PD       0:00      1 (Resources)
             40070       gpu   T_SP_7  akashpg PD       0:00      1 (Priority)
             40071       gpu     T_29 pmanikan PD       0:00      1 (Priority)
             40268       gpu     test m22cs053 PD       0:00      1 (Priority)
             40552       gpu b19cse03 b19cse03 PD       0:00      1 (Priority)
             40605       gpu b19cse03 b19cse03 PD       0:00      1 (Priority)
             40620       gpu    lab56 m22cs056 PD       0:00      1 (Priority)
             40687       gpu    lab56 m22cs056 PD       0:00      1 (Priority)
             40702       gpu    lab56 m22cs056 PD       0:00      1 (Priority)
             40990       gpu myjob104   jethi1 PD       0:00      1 (Priority)
             41395       gpu loptical   swamy2 PD       0:00      1 (Priority)
             38900       gpu    ec0.1  santosh  R 2-10:03:47      1 gpu1
             39144       gpu   gpu-cr    alam2  R 2-10:03:47      1 gpu1
             39145       gpu   gpu-cu    alam2  R 2-10:03:47      1 gpu1
             39148       gpu   gpu-mo    alam2  R 2-10:03:47      1 gpu1
             39307       gpu  101-110  akashpg  R 2-10:03:47      1 gpu1
             39440       gpu  T_SP_18 saptarsh  R 2-10:03:47      1 gpu1
             39521       gpu     T_26 pmanikan  R 2-10:03:47      1 gpu1
             39758       gpu  111-120  akashpg  R 2-10:03:47      1 gpu1
             39978       gpu    SnBr2 satyajit  R 2-10:03:47      1 gpu1
             39989       gpu  T_SP_16 saptarsh  R 2-10:03:47      1 gpu1
             40001       gpu       30 patanvad  R 2-10:03:47      1 gpu1
             40003       gpu       40 patanvad  R 2-10:03:47      1 gpu1
             40038       gpu    SrBr2 satyajit  R 2-10:03:19      1 gpu1
             40726       gpu unit-opt meghnani  R 1-23:19:22      1 gpu1
             41078       gpu       ru   swamy2  R 1-00:11:15      1 gpu1
             41354       gpu    large   swamy2  R   11:08:18      1 gpu1
             41387      gpu2 dlops3Jo b20ai058  R    2:28:05      1 gpu2
             41408      gpu2 b20bb025 b20bb025  R      25:35      1 gpu2
             41411      gpu2 test_job b20ai052  R       0:35      1 gpu2
             40825     small   280_10 ananya_r PD       0:00      1 (Resources)
             40966     small test_job b20bb008 PD       0:00      1 (Priority)
             41283     small    Trail abhinand PD       0:00      1 (Priority)
             41394     small    inter amitava_ PD       0:00      1 (Priority)
             39574     small   290_10 ananya_r  R 2-01:36:49      1 cn01
             40026     small      us2 dhanger1  R 4-04:06:23      1 cn02
             40028     small      us3 dhanger1  R 4-04:00:52      1 cn02
             40518     small     job0  ansari1  R 1-17:23:46      1 cn02
             40822     small   270_10 ananya_r  R   18:02:17      1 cn03
             41060      test     t7_4 m22cy007  R 1-02:07:29      1 cn20
             41366      test t8_300sm  mahesh2  R    4:33:09      1 cn20
             41398      test    pdisp  sindhu1  R    1:27:10      1 cn20
```

- The directory in the server looked like (after ls):

```
[b20ai052@hpclogin Lab 7 Slurm]$ ls
B20AI052_Lab_Assignment_7.py  B20AI052_Lab_Assignment_7.sh  data  images  models
```

- Then, submitted the job using the command:
    - sbatch B20AI052_Lab_Assignment_7.sh
- Got the Job-id as:

```
[b20ai052@hpclogin Lab 7 Slurm]$ sbatch B20AI052_Lab_Assignment_7.sh
Submitted batch job 41411
```

- Then, my slurm log was created in some time:

```
[b20ai052@hpclogin Lab 7 Slurm]$ ls
B20AI052_Lab_Assignment_7.py  B20AI052_Lab_Assignment_7.sh  data  images  models  slurm-41411.out
```

- Used the command:
    - tail -f slurm-41411.out for looking at the output contents.
- Output generated was as follows:

```
[b20ai052@hpclogin Lab 7 Slurm]$ tail -f slurm-41411.out
6144it [00:00, 14198239.00it/s]         Extracting ./data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to ./data/FashionMNIST/raw

Dataset downloaded!
cuda:0
Train data shape:  torch.Size([60000, 28, 28])
Test data shape:  torch.Size([10000, 28, 28])
Train labels shape:  torch.Size([60000])
Test labels shape:  torch.Size([10000])
Training the model for 10 epochs

Epoch: 1, Loss: 0.3546, Acc: 0.8710
Epoch: 2, Loss: 0.2561, Acc: 0.9073
Epoch: 3, Loss: 0.2191, Acc: 0.9203
Epoch: 4, Loss: 0.1981, Acc: 0.9281
Epoch: 5, Loss: 0.1793, Acc: 0.9353
Epoch: 6, Loss: 0.1670, Acc: 0.9398
Epoch: 7, Loss: 0.1538, Acc: 0.9439
Epoch: 8, Loss: 0.1397, Acc: 0.9489
Epoch: 9, Loss: 0.1304, Acc: 0.9534
Epoch: 10, Loss: 0.1185, Acc: 0.9561
Time taken for 10 epochs:  690.7021113457158
Test set: Average loss: 0.0031, Accuracy: 9300/10000 (93%)
Training the model for 15 epochs
Epoch: 1, Loss: 0.3582, Acc: 0.8721
Epoch: 2, Loss: 0.2565, Acc: 0.9077
Epoch: 3, Loss: 0.2201, Acc: 0.9196
Epoch: 4, Loss: 0.2000, Acc: 0.9269
Epoch: 5, Loss: 0.1815, Acc: 0.9334
Epoch: 6, Loss: 0.1682, Acc: 0.9388
Epoch: 7, Loss: 0.1541, Acc: 0.9436
Epoch: 8, Loss: 0.1425, Acc: 0.9479
Epoch: 9, Loss: 0.1302, Acc: 0.9531
Epoch: 10, Loss: 0.1231, Acc: 0.9554
Epoch: 11, Loss: 0.1096, Acc: 0.9600
Epoch: 12, Loss: 0.0994, Acc: 0.9638
Epoch: 13, Loss: 0.0917, Acc: 0.9662
Epoch: 14, Loss: 0.0839, Acc: 0.9698
Epoch: 15, Loss: 0.0733, Acc: 0.9728
Time taken for 15 epochs:  1133.5947534926236
Test set: Average loss: 0.0035, Accuracy: 9364/10000 (94%)
```

- Then, output for hyperparameter tuning:

```
Epoch: 1, Loss: 0.4257, Acc: 0.8573
Epoch: 2, Loss: 0.2412, Acc: 0.9156
Epoch: 3, Loss: 0.2035, Acc: 0.9282
Epoch: 4, Loss: 0.1813, Acc: 0.9352
Epoch: 5, Loss: 0.1652, Acc: 0.9418
Epoch: 6, Loss: 0.1487, Acc: 0.9470
Epoch: 7, Loss: 0.1382, Acc: 0.9507
Epoch: 8, Loss: 0.1274, Acc: 0.9546
Epoch: 9, Loss: 0.1172, Acc: 0.9581
Epoch: 10, Loss: 0.1069, Acc: 0.9623
Epoch: 1, Loss: 0.4301, Acc: 0.8546
Epoch: 2, Loss: 0.2458, Acc: 0.9123
Epoch: 3, Loss: 0.2086, Acc: 0.9259
Epoch: 4, Loss: 0.1855, Acc: 0.9343
Epoch: 5, Loss: 0.1662, Acc: 0.9412
Epoch: 6, Loss: 0.1532, Acc: 0.9459
Epoch: 7, Loss: 0.1392, Acc: 0.9499
Epoch: 8, Loss: 0.1306, Acc: 0.9536
Epoch: 9, Loss: 0.1155, Acc: 0.9588
Epoch: 10, Loss: 0.1091, Acc: 0.9623
Epoch: 1, Loss: 0.4228, Acc: 0.8582
Epoch: 2, Loss: 0.2445, Acc: 0.9144
Epoch: 3, Loss: 0.2101, Acc: 0.9259
Epoch: 4, Loss: 0.1892, Acc: 0.9352
Epoch: 5, Loss: 0.1732, Acc: 0.9402
Epoch: 6, Loss: 0.1634, Acc: 0.9439
Epoch: 7, Loss: 0.1561, Acc: 0.9468
Epoch: 8, Loss: 0.1484, Acc: 0.9498
Epoch: 9, Loss: 0.1435, Acc: 0.9520
Epoch: 10, Loss: 0.1380, Acc: 0.9547
Epoch: 1, Loss: 0.3969, Acc: 0.8613
Epoch: 2, Loss: 0.2202, Acc: 0.9211
Epoch: 3, Loss: 0.1855, Acc: 0.9321
Epoch: 4, Loss: 0.1637, Acc: 0.9395
Epoch: 5, Loss: 0.1485, Acc: 0.9458
Epoch: 6, Loss: 0.1342, Acc: 0.9501
Epoch: 7, Loss: 0.1213, Acc: 0.9550
Epoch: 8, Loss: 0.1171, Acc: 0.9569
Epoch: 9, Loss: 0.0995, Acc: 0.9639
Epoch: 10, Loss: 0.0960, Acc: 0.9656
Epoch: 1, Loss: 0.3927, Acc: 0.8612
Epoch: 2, Loss: 0.2223, Acc: 0.9193
Epoch: 3, Loss: 0.1913, Acc: 0.9305
Epoch: 4, Loss: 0.1747, Acc: 0.9362
Epoch: 5, Loss: 0.1596, Acc: 0.9419
Epoch: 6, Loss: 0.1554, Acc: 0.9441
Epoch: 7, Loss: 0.1563, Acc: 0.9427
```

- Trained 27 different models using all the combinations of the following parameters:
    - learning_rates = [0.001, 0.0001, 0.00001]
    - momentums = [0.9, 0.99, 0.999]
    - weight_decays = [0.0001, 0.001, 0.01]
- Also, in my server directory folders named data, models, images were created where all the data, trained models and generated plots were saved respectively after the code compilation process.
- Then, after hyperparameter tuning got the results for the best model:

```
Best model:  lr                0.000100
momentum          0.990000
weight_decay      0.001000
train_loss        0.108368
train_acc         0.962550
test_loss         0.002749
test_acc          0.942100
Name: 13, dtype: float64
Best accuracy:  0.9421
Learning rate:  0.0001
Momentum:  0.99
Weight decay:  0.001
```