

# Handwritten Digit Recognition

## Initial Setup

- I entered AWS through the gateway in the course and open SageMaker Studio. Downloaded and made the dataset available.
- Started by installing all the requirements in the jupyter notebook "mnist-handwritten-digits-classification-experiment (1).ipynb."
- I used the basic "Python 3 (Data science - 01)" Kernel in this project. Also essential PyTorch libraries have been installed.

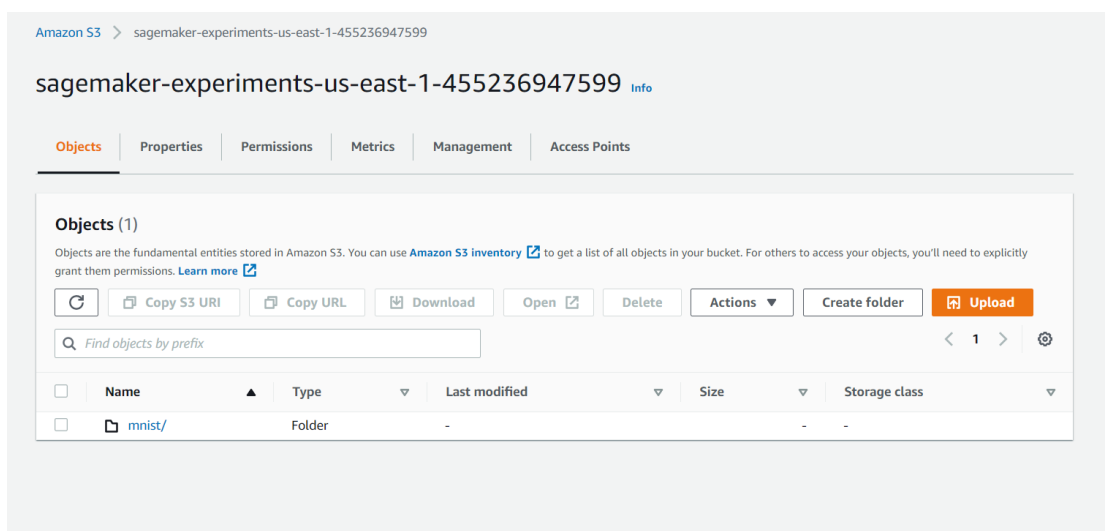
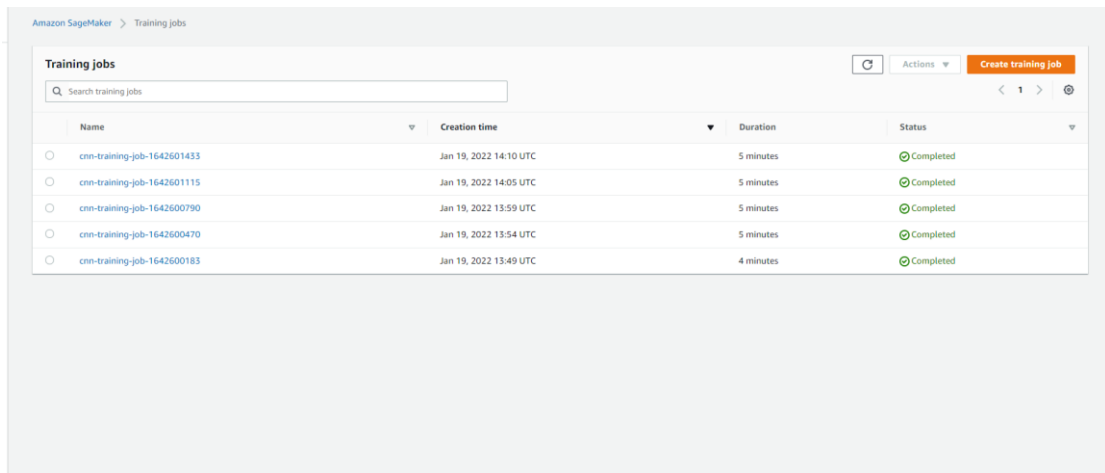


Figure 1: S3 Bucket Setup

- The project run on SageMaker, such as Autopilot jobs and training jobs, will be logged automatically. You may also keep track of artefacts for other phases in an ML workflow that occur before or after model training, such as data pre-processing or model assessment.
- I trained a Convolutional Neural Network (CNN) model for this project. Adjust the number of hidden channels in the model by adjusting the hyperparameter. Using SageMaker Experiments, make track of parameter settings and model accuracy.
- I focused on various settings for the number of hidden channels in the CNN model while training it on SageMaker. We'll set up a Trial to keep track of each training task. I also made a Trial Component out of the tracker we made before and include it in the Trial. This will add the parameters we collected during the data pre-processing step to the Trial.
- I used 5 hidden channels i.e 2, 5, 10, 20, 32 to train the model
- Each Training job defines the individual hidden channel for training.



The screenshot shows the Amazon SageMaker console's 'Training jobs' page. At the top, there's a search bar and a 'Create training job' button. Below is a table with columns: Name, Creation time, Duration, and Status. Five training jobs are listed, all with a status of 'completed'.

| Name                        | Creation time          | Duration  | Status    |
|-----------------------------|------------------------|-----------|-----------|
| cnn-training-job-1642601433 | Jan 19, 2022 14:10 UTC | 5 minutes | completed |
| cnn-training-job-1642601115 | Jan 19, 2022 14:05 UTC | 5 minutes | completed |
| cnn-training-job-1642600790 | Jan 19, 2022 13:59 UTC | 5 minutes | completed |
| cnn-training-job-1642600470 | Jan 19, 2022 13:54 UTC | 5 minutes | completed |
| cnn-training-job-1642600183 | Jan 19, 2022 13:49 UTC | 4 minutes | completed |

Figure 2: Training jobs

## Training

- Training using python script is performed. Script mode for PyTorch is a training script format that allows you to run any PyTorch training script in SageMaker with few changes. Transferring your script to a SageMaker training instance is handled by the SageMaker Python SDK. SageMaker's native PyTorch support on the training instance sets up training-related environment variables and runs your training script. I utilised the SageMaker Python SDK to initiate a training job and deploy the trained model in this tutorial.

```

Based on https://github.com/pytorch/examples/blob/master/mnist/main.py
class Net(nn.Module):
    def __init__(self, hidden_channels, kernel_size, drop_out):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, hidden_channels, kernel_size=kernel_size)
        self.conv2 = nn.Conv2d(hidden_channels, 20, kernel_size=kernel_size)
        self.conv2_drop = nn.Dropout2d(p=drop_out)
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)

    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x), 2))
        x = F.relu(F.max_pool2d(self.conv2_drop(self.conv2(x)), 2))
        x = x.view(-1, 320)
        x = F.relu(self.fc1(x))
        x = F.dropout(x, training=self.training)
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)

def _get_train_data_loader(batch_size, training_dir, is_distributed, **kwargs):
    logger.info("Get train data loader")
    dataset = datasets.MNIST(
        training_dir,
        train=True,
        transform=transforms.Compose(
            [transforms.ToTensor(), transforms.Normalize((0.1307,), (0.3081,))]
        ),
        download=False,
    )
    train_sampler = (
        torch.utils.data.distributed.DistributedSampler(dataset) if is_distributed else None
    )
    return torch.utils.data.DataLoader(
        dataset,
        batch_size=batch_size,

```

Figure 3: Python script

- The mnist.py script contains all of the code required to train and host a SageMaker model (the model fn function is used to load a model). The training script is quite similar to a training script that you could run outside of SageMaker, however you may access important training environment attributes through several environment variables, such as:

**SM\_MODEL\_DIR:** A string representing the path to the directory to write model artifacts to. These artifacts are uploaded to S3 for model hosting.

**SM\_CURRENT\_HOST:** The name of the current container on the container network.

**SM\_HOSTS:** JSON encoded list containing all the hosts.

**SM\_CHANNEL\_TRAINING:** A string representing the path to the directory containing data in the 'training' channel.

- The PyTorch class enables us to run our training function on SageMaker infrastructure as a training task. I needed training script, an IAM role, the number of training instances, the kind of training instance, and hyperparameters to set it up. I executed training task on `ml.m5.xlarge` instance in this scenario.

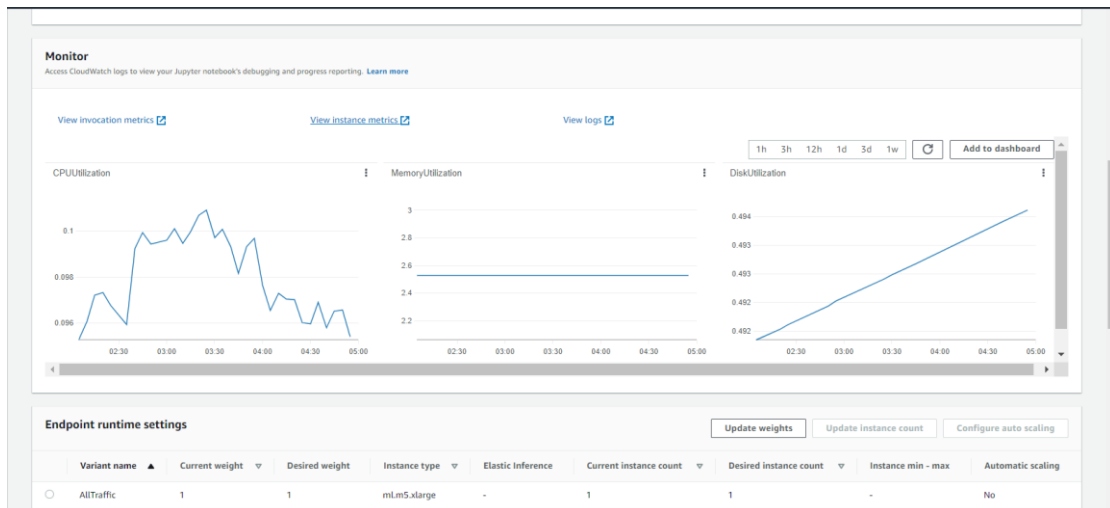


Figure 4: Endpoint monitor on `ml.m5.xlarge` instance

- The deployed endpoint for the training job used for prediction.

Amazon SageMaker > Endpoints

Endpoints

[Update endpoint](#) [Actions](#) [Create endpoint](#)

| Name  | ARN   | Creation time          | Status    | Last updated           |
|---|---|------------------------|-----------|------------------------|
| crn-training-job-1642600183-aws-trainin-2022-01-19-14-17-52-676 | arn:aws:sagemaker:us-east-1:455236947599:endpoint/crn-training-job-1642600183-aws-trainin-2022-01-19-14-17-52-676 | Jan 19, 2022 14:17 UTC | InService | Jan 19, 2022 14:19 UTC |

Figure 5: Endpoint

## Results:

Submitted by  
Mansi Chilwant

- The summary of Training and endpoint along side with CPU usage is provided in profiler report.
- Using several hidden channels, I was able to achieve ~ 97 percent accuracy for handwritten digit identification for two epochs.
- The aim of the project was to deliver the highest accuracy for handwritten digit recognition system using Pytorch model. The tabular data confirms the highest accuracy using Pytorch model.

```

2022-01-19T19:23:19.601+05:30 SH_HP_BACKEND=gloo
2022-01-19T19:23:19.601+05:30 SH_HP_EPOCHS=2
2022-01-19T19:23:19.601+05:30 SH_HP_KERNEL_SIZE=5
2022-01-19T19:23:19.601+05:30 PYTHONPATH=/usr/local/bin:/usr/lib/python3.6:/usr/lib/python3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/dist-packages
2022-01-19T19:23:19.601+05:30 Invoking script with the following command:
2022-01-19T19:23:19.601+05:30 /usr/bin/python -m mnist --backend gloo --dropout 0.2 --epochs 2 --hidden_channels 2 --kernel_size 5 --optimizer sgd
2022-01-19T19:23:11.602+05:30 Distributed training - False
2022-01-19T19:23:11.602+05:30 Number of gpus available - 0
2022-01-19T19:23:11.602+05:30 Get train data loader
2022-01-19T19:23:11.602+05:30 Get test data loader
2022-01-19T19:23:11.602+05:30 Processes 60000/60000 (100%) of train data
2022-01-19T19:23:11.602+05:30 Processes 10000/10000 (100%) of test data
2022-01-19T19:23:11.603+05:30 Train Epoch: 1 [6400/60000 (11%)], Train Loss: 1.617049;
2022-01-19T19:23:16.604+05:30 Train Epoch: 1 [12800/60000 (21%)], Train Loss: 0.943270;
2022-01-19T19:23:18.605+05:30 Train Epoch: 1 [19200/60000 (32%)], Train Loss: 0.643992;
2022-01-19T19:23:20.605+05:30 Train Epoch: 1 [25600/60000 (43%)], Train Loss: 0.432059;
2022-01-19T19:23:22.606+05:30 Train Epoch: 1 [32000/60000 (53%)], Train Loss: 0.464780;
2022-01-19T19:23:23.607+05:30 Train Epoch: 1 [38400/60000 (64%)], Train Loss: 0.322854;
2022-01-19T19:23:25.607+05:30 Train Epoch: 1 [44800/60000 (75%)], Train Loss: 0.351526;
2022-01-19T19:23:27.609+05:30 Train Epoch: 1 [51200/60000 (85%)], Train Loss: 0.389387;
2022-01-19T19:23:29.609+05:30 Train Epoch: 1 [57600/60000 (96%)], Train Loss: 0.375708;
2022-01-19T19:23:31.691+05:30 Test Average Loss: 0.1892, Test Accuracy: 95%
2022-01-19T19:23:34.692+05:30 Train Epoch: 2 [6400/60000 (11%)], Train Loss: 0.275796;
2022-01-19T19:23:36.693+05:30 Train Epoch: 2 [12800/60000 (21%)], Train Loss: 0.292543;
2022-01-19T19:23:38.694+05:30 Train Epoch: 2 [19200/60000 (32%)], Train Loss: 0.244605;
2022-01-19T19:23:40.695+05:30 Train Epoch: 2 [25600/60000 (43%)], Train Loss: 0.203167;
2022-01-19T19:23:42.696+05:30 Train Epoch: 2 [32000/60000 (53%)], Train Loss: 0.279576;
2022-01-19T19:23:43.696+05:30 Train Epoch: 2 [38400/60000 (64%)], Train Loss: 0.343496;
2022-01-19T19:23:45.697+05:30 Train Epoch: 2 [44800/60000 (75%)], Train Loss: 0.414407;
2022-01-19T19:23:47.698+05:30 Train Epoch: 2 [51200/60000 (85%)], Train Loss: 0.393495;
2022-01-19T19:23:49.699+05:30 Train Epoch: 2 [57600/60000 (96%)], Train Loss: 0.157259;
2022-01-19T19:23:52.700+05:30 Test Average Loss: 0.1156, Test Accuracy: 97%
2022-01-19T19:23:52.700+05:30 Saving the model.
2022-01-19T19:23:52.700+05:30 2022-01-19 13:53:51,669 sagemaker-containers zero Reporting training success
No newer events at this moment. Auto retry paused. Resume

```

Figure 6: Training Job 1: Hidden Channel: 2, Test Accuracy of min.95% max 97%

```
2022-01-19T19:10:13.220+05:30 SH_HP_BACKEND=gloo
2022-01-19T19:10:13.220+05:30 SH_HP_EPISODE=2
2022-01-19T19:10:13.220+05:30 SH_HP_KERNEL_SIZE=5
2022-01-19T19:10:13.220+05:30 PYTHONPATH=/usr/local/bin:/usr/lib/python3.6:/usr/lib/python3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/dist-packages
2022-01-19T19:10:13.220+05:30 Invoking script with the following command:
2022-01-19T19:10:13.220+05:30 /usr/bin/python -m mist --backend gloo --dropout 0.2 --epochs 2 --hidden_channels 5 --kernel_size 5 --optimizer sgd
2022-01-19T19:10:14.221+05:30 Distributed training - False
2022-01-19T19:10:14.221+05:30 Number of gpus available - 0
2022-01-19T19:10:14.221+05:30 Get train data loader
2022-01-19T19:10:14.221+05:30 Get test data loader
2022-01-19T19:10:14.221+05:30 Processes 60000/60000 (100%) of train data
2022-01-19T19:10:14.221+05:30 Processes 10000/10000 (100%) of test data
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [6400/60000 (11%)], Train Loss: 1.696200;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [12800/60000 (21%)], Train Loss: 0.994831;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [19200/60000 (32%)], Train Loss: 0.615339;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [25600/60000 (43%)], Train Loss: 0.648925;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [32000/60000 (53%)], Train Loss: 0.571486;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [38400/60000 (64%)], Train Loss: 0.793993;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [44800/60000 (75%)], Train Loss: 0.430899;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [51200/60000 (85%)], Train Loss: 0.549112;
2022-01-19T19:10:16.222+05:30 Train Epoch: 1 [57600/60000 (96%)], Train Loss: 0.480673;
2022-01-19T19:10:16.222+05:30 Test Average Loss: 0.1914, Test Accuracy: 94%;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [6400/60000 (11%)], Train Loss: 0.297187;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [12800/60000 (21%)], Train Loss: 0.364007;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [19200/60000 (32%)], Train Loss: 0.368093;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [25600/60000 (43%)], Train Loss: 0.339477;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [32000/60000 (53%)], Train Loss: 0.367964;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [38400/60000 (64%)], Train Loss: 0.402320;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [44800/60000 (75%)], Train Loss: 0.203390;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [51200/60000 (85%)], Train Loss: 0.445356;
2022-01-19T19:10:16.222+05:30 Train Epoch: 2 [57600/60000 (96%)], Train Loss: 0.197944;
2022-01-19T19:10:16.222+05:30 Test Average Loss: 0.1157, Test Accuracy: 96%;
2022-01-19T19:10:16.222+05:30 Saving the model.
2022-01-19T19:10:16.222+05:30 2022-01-19 13:56:54,809 sagemaker-containers INFO Reporting training SUCCESS
```

Figure 7: Training Job 2: Hidden Channel: 5, Test Accuracy of min.94% max 96%

```
2022-01-19T19:10:56.609+05:30 SH_HP_BACKEND=gloo
2022-01-19T19:10:56.609+05:30 SH_HP_EPISODE=2
2022-01-19T19:10:56.609+05:30 SH_HP_KERNEL_SIZE=5
2022-01-19T19:10:56.609+05:30 PYTHONPATH=/usr/local/bin:/usr/lib/python3.6:/usr/lib/python3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/dist-packages
2022-01-19T19:10:56.609+05:30 Invoking script with the following command:
2022-01-19T19:10:56.609+05:30 /usr/bin/python -m mist --backend gloo --dropout 0.2 --epochs 2 --hidden_channels 10 --kernel_size 5 --optimizer sgd
2022-01-19T19:10:56.609+05:30 Distributed training - False
2022-01-19T19:10:56.609+05:30 Number of gpus available - 0
2022-01-19T19:10:56.609+05:30 Get train data loader
2022-01-19T19:10:56.609+05:30 Get test data loader
2022-01-19T19:10:56.609+05:30 Processes 60000/60000 (100%) of train data
2022-01-19T19:10:56.609+05:30 Processes 10000/10000 (100%) of test data
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [6400/60000 (11%)], Train Loss: 1.695285;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [12800/60000 (21%)], Train Loss: 0.928432;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [19200/60000 (32%)], Train Loss: 0.782160;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [25600/60000 (43%)], Train Loss: 0.442373;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [32000/60000 (53%)], Train Loss: 0.413047;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [38400/60000 (64%)], Train Loss: 0.501332;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [44800/60000 (75%)], Train Loss: 0.303585;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [51200/60000 (85%)], Train Loss: 0.328480;
2022-01-19T19:10:58.610+05:30 Train Epoch: 1 [57600/60000 (96%)], Train Loss: 0.396889;
2022-01-19T19:10:58.610+05:30 Test Average Loss: 0.1679, Test Accuracy: 95%;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [6400/60000 (11%)], Train Loss: 0.603319;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [12800/60000 (21%)], Train Loss: 0.229334;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [19200/60000 (32%)], Train Loss: 0.201790;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [25600/60000 (43%)], Train Loss: 0.376957;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [32000/60000 (53%)], Train Loss: 0.412061;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [38400/60000 (64%)], Train Loss: 0.200818;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [44800/60000 (75%)], Train Loss: 0.233049;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [51200/60000 (85%)], Train Loss: 0.319483;
2022-01-19T19:10:58.610+05:30 Train Epoch: 2 [57600/60000 (96%)], Train Loss: 0.210038;
2022-01-19T19:10:58.610+05:30 Test Average Loss: 0.1068, Test Accuracy: 97%;
2022-01-19T19:10:58.610+05:30 Saving the model.
2022-01-19T19:10:58.610+05:30 2022-01-19 14:04:18,603 sagemaker-containers INFO Reporting training SUCCESS
```

Figure 8: Training Job 3: Hidden Channel: 10, Test Accuracy of min.95% max 97%

```

services, features, blogs, docs, and more [Alt+S]
2022-01-19T19:38:56.874+05:30 DL_HP_BACKEND=gloo
2022-01-19T19:38:56.874+05:30 DL_HP_EPOCHS=2
2022-01-19T19:38:56.874+05:30 DL_HP_KERNEL_SIZE=5
2022-01-19T19:38:56.874+05:30 PYTHONPATH=/usr/local/bin:/usr/lib/python3.11p:/usr/lib/python3.6:/usr/lib/python3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/dist-packages
2022-01-19T19:38:56.874+05:30 Invoking script with the following command:
2022-01-19T19:38:56.874+05:30 /usr/bin/python -m mnist --backend gloo --dropout 0.2 --epochs 2 --hidden_channels 20 --kernel_size 5 --optimizer sgd
2022-01-19T19:38:57.874+05:30 Distributed training - False
2022-01-19T19:38:57.874+05:30 Number of gpus available - 0
2022-01-19T19:38:57.874+05:30 Get train data loader
2022-01-19T19:38:57.874+05:30 Get test data loader
2022-01-19T19:38:57.874+05:30 Processes 60000/60000 (100%) of train data
2022-01-19T19:38:57.874+05:30 Processes 10000/10000 (100%) of test data
2022-01-19T19:39:00.888+05:30 Train Epoch: 1 [6400/60000 (11%)], Train Loss: 1.821211
2022-01-19T19:39:04.888+05:30 Train Epoch: 1 [12800/60000 (21%)], Train Loss: 1.813725
2022-01-19T19:39:07.889+05:30 Train Epoch: 1 [19200/60000 (32%)], Train Loss: 0.799500
2022-01-19T19:39:10.890+05:30 Train Epoch: 1 [25600/60000 (43%)], Train Loss: 0.595724
2022-01-19T19:39:13.891+05:30 Train Epoch: 1 [32000/60000 (53%)], Train Loss: 0.411506
2022-01-19T19:39:16.892+05:30 Train Epoch: 1 [38400/60000 (64%)], Train Loss: 0.376129
2022-01-19T19:39:19.893+05:30 Train Epoch: 1 [44800/60000 (75%)], Train Loss: 0.317877
2022-01-19T19:39:22.894+05:30 Train Epoch: 1 [51200/60000 (85%)], Train Loss: 0.293233
2022-01-19T19:39:24.895+05:30 Train Epoch: 1 [57600/60000 (96%)], Train Loss: 0.204020
2022-01-19T19:39:26.896+05:30 Test Average Loss: 0.3407, Test Accuracy: 96%
2022-01-19T19:39:31.897+05:30 Train Epoch: 2 [6400/60000 (11%)], Train Loss: 0.384839
2022-01-19T19:39:33.898+05:30 Train Epoch: 2 [12800/60000 (21%)], Train Loss: 0.280609
2022-01-19T19:39:36.900+05:30 Train Epoch: 2 [19200/60000 (32%)], Train Loss: 0.299078
2022-01-19T19:39:39.901+05:30 Train Epoch: 2 [25600/60000 (43%)], Train Loss: 0.451251
2022-01-19T19:39:42.904+05:30 Train Epoch: 2 [32000/60000 (53%)], Train Loss: 0.263621
2022-01-19T19:39:44.907+05:30 Train Epoch: 2 [38400/60000 (64%)], Train Loss: 0.251691
2022-01-19T19:39:47.908+05:30 Train Epoch: 2 [44800/60000 (75%)], Train Loss: 0.375436
2022-01-19T19:39:50.909+05:30 Train Epoch: 2 [51200/60000 (85%)], Train Loss: 0.332934
2022-01-19T19:39:52.910+05:30 Train Epoch: 2 [57600/60000 (96%)], Train Loss: 0.227200
2022-01-19T19:39:56.911+05:30 Test Average Loss: 0.4965, Test Accuracy: 97%
2022-01-19T19:39:56.911+05:30 Saving the model.

```

Figure 9: Training Job 4: Hidden Channel: 20, Test Accuracy of min.96% max 97%

```

2022-01-19T19:44:23.182+05:30 DL_HP_BACKEND=gloo
2022-01-19T19:44:23.182+05:30 DL_HP_EPOCHS=2
2022-01-19T19:44:23.182+05:30 DL_HP_KERNEL_SIZE=5
2022-01-19T19:44:23.182+05:30 PYTHONPATH=/usr/local/bin:/usr/lib/python3.11p:/usr/lib/python3.6:/usr/lib/python3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/dist-packages
2022-01-19T19:44:23.182+05:30 Invoking script with the following command:
2022-01-19T19:44:23.182+05:30 /usr/bin/python -m mnist --backend gloo --dropout 0.2 --epochs 2 --hidden_channels 32 --kernel_size 5 --optimizer sgd
2022-01-19T19:44:23.182+05:30 Distributed training - False
2022-01-19T19:44:23.182+05:30 Number of gpus available - 0
2022-01-19T19:44:23.182+05:30 Get train data loader
2022-01-19T19:44:23.182+05:30 Get test data loader
2022-01-19T19:44:23.182+05:30 Processes 60000/60000 (100%) of train data
2022-01-19T19:44:23.182+05:30 Processes 10000/10000 (100%) of test data
2022-01-19T19:44:26.183+05:30 Train Epoch: 1 [6400/60000 (11%)], Train Loss: 1.462234
2022-01-19T19:44:29.184+05:30 Train Epoch: 1 [12800/60000 (21%)], Train Loss: 0.257541
2022-01-19T19:44:31.185+05:30 Train Epoch: 1 [19200/60000 (32%)], Train Loss: 0.375466
2022-01-19T19:44:34.186+05:30 Train Epoch: 1 [25600/60000 (43%)], Train Loss: 0.530801
2022-01-19T19:44:37.187+05:30 Train Epoch: 1 [32000/60000 (53%)], Train Loss: 0.445891
2022-01-19T19:44:40.188+05:30 Train Epoch: 1 [38400/60000 (64%)], Train Loss: 0.388932
2022-01-19T19:44:43.189+05:30 Train Epoch: 1 [44800/60000 (75%)], Train Loss: 0.451403
2022-01-19T19:44:46.190+05:30 Train Epoch: 1 [51200/60000 (85%)], Train Loss: 0.382861
2022-01-19T19:44:49.191+05:30 Train Epoch: 1 [57600/60000 (96%)], Train Loss: 0.379461
2022-01-19T19:44:51.192+05:30 Test Average Loss: 0.551, Test Accuracy: 95%
2022-01-19T19:44:56.193+05:30 Train Epoch: 2 [6400/60000 (11%)], Train Loss: 0.382234
2022-01-19T19:45:00.194+05:30 Train Epoch: 2 [12800/60000 (21%)], Train Loss: 0.531221
2022-01-19T19:45:03.195+05:30 Train Epoch: 2 [19200/60000 (32%)], Train Loss: 0.399507
2022-01-19T19:45:06.196+05:30 Train Epoch: 2 [25600/60000 (43%)], Train Loss: 0.388622
2022-01-19T19:45:09.197+05:30 Train Epoch: 2 [32000/60000 (53%)], Train Loss: 0.377258
2022-01-19T19:45:12.198+05:30 Train Epoch: 2 [38400/60000 (64%)], Train Loss: 0.227790
2022-01-19T19:45:15.199+05:30 Train Epoch: 2 [44800/60000 (75%)], Train Loss: 0.342807
2022-01-19T19:45:18.200+05:30 Train Epoch: 2 [51200/60000 (85%)], Train Loss: 0.385124
2022-01-19T19:45:21.201+05:30 Train Epoch: 2 [57600/60000 (96%)], Train Loss: 0.393494
2022-01-19T19:45:23.202+05:30 Test Average Loss: 0.5812, Test Accuracy: 97%
2022-01-19T19:45:23.202+05:30 Saving the model.
2022-01-19 14:31:18,917 Engineer-Containers DPO Reporting Training Success

```

Figure 10: Training Job 5: Hidden Channel: 32, Test Accuracy of min.95% max 97%