ASL-alphabets Prediction using Transfer Learning on AWS SageMaker

This notebook walks through implementation of Image Classification Machine Learning Model to alphabets based on ASL-alphabets to translate ASL gesture to English

- We have used data from Kaggle, link is mentioned below: https://www.kaggle.com/datasets/shadman0786/asl-alphabet-cnn (https://www.kaggle.com/datasets/shadman0786/asl-alphabet-cnn)
- We will be using a pretrained Resnet50 model from pytorch vision library (https://pytorch.org/vision/master/generated/torchvision.models.resnet50.html (https://pytorch.org/vision/master/generated/torchvision.models.resnet50.html))
- We will add two Fully connected Neural Network layers on top of the above Resnet50 model
- We will use concept of Transfer learning therefore we will be freezing all the existing Convolutional layers in the pretrained Resnet50 model and only change the gradients for the two fully connected layers
- We perform Hyperparameter tuning, to get the optimal best hyperparameters to be used in our model
- We have added configuration for Profiling and Debugging our training model by adding relevant hooks in Training and Testing (evel) phases
- We will then deploy our Model, for deploying we have created inference script. The inference script will be overriding a few functions that will be used by our deployed endpoint for making inferences/predictions.
- At the end, we would be testing our model with some test images of dogs, to verify if the model is working as per our expectations.

In [2]:

```
# Install any packages that you might need
# need the smdebug package
!pip install smdebug
```

```
Keyring is skipped due to an exception: 'keyring.backends'
Collecting smdebug
  Using cached smdebug-1.0.12-py2.py3-none-any.whl (270 kB)
Collecting pyinstrument==3.4.2
  Using cached pyinstrument-3.4.2-py2.py3-none-any.whl (83 kB)
Requirement already satisfied: boto3>=1.10.32 in /opt/conda/lib/python3.7/site-packages (from smdebug) (1.26.24)
Requirement already satisfied: numpy>=1.16.0 in /opt/conda/lib/python3.7/site-packages (from smdebug) (1.21.6)
Requirement already satisfied: protobuf>=3.6.0 in /opt/conda/lib/python3.7/site-packages (from smdebug) (3.20.3)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-packages (from smdebug) (20.1)
Collecting pyinstrument-cext>=0.2.2
  Using cached pyinstrument_cext-0.2.4-cp37-cp37m-manylinux2010_x86_64.whl (20 kB)
Requirement already satisfied: botocore<1.30.0,>=1.29.24 in /opt/conda/lib/python3.7/site-packages (from boto3>=1.10.32->smdebug) (1.29.
24)
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in /opt/conda/lib/python3.7/site-packages (from boto3>=1.10.32->smdebug) (0.6.0)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /opt/conda/lib/python3.7/site-packages (from boto3>=1.10.32->smdebug) (1.0.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from packaging->smdebug) (1.14.0)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (from packaging->smdebug) (2.4.6)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/lib/python3.7/site-packages (from botocore<1.30.0,>=1.29.24->bo
to3>=1.10.32->smdebug) (2.8.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /opt/conda/lib/python3.7/site-packages (from botocore<1.30.0,>=1.29.24->boto3>=
1.10.32->smdebug) (1.26.13)
Installing collected packages: pyinstrument-cext, pyinstrument, smdebug
Successfully installed pyinstrument-3.4.2 pyinstrument-cext-0.2.4 smdebug-1.0.12
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It i
s recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip available: 22.3.1 -> 23.0
[notice] To update, run: pip install --upgrade pip
```

In [3]:

```
# TODO: Import any packages that you might need
# For instance you will need Boto3 and Sagemaker
import sagemaker
import boto3
from sagemaker.session import Session
from sagemaker import get_execution_role
# Initializing some useful variables
role = get_execution_role()
sagemaker_session = sagemaker.Session()
region = sagemaker_session.boto_region_name
bucket = sagemaker_session.default_bucket()
print(f"Region {region}")
print(f"Default s3 bucket : {bucket}")
```

```
Region us-east-1
Default s3 bucket : sagemaker-us-east-1-910293992207
```

Dataset

The dataset we used for this project is ASL-alphabet dataset that can be found in the [link]https://www.kaggle.com/datasets/shadman0786/asl-alphabet-cnn (https://www.kaggle.com/datasets/shadman0786/asl-alphabet-cnn)

It comprises: -asl_alphabet

- Test
  -- A (600)
  -- B (600)

  ....
  -- Z (600)

- Train folder

  -- A (2400)

  -- B (2400)

  ....

  -- Z (2400)

In [4]:
```
!pip install kaggle
```

```
Keyring is skipped due to an exception: 'keyring.backends'
Collecting kaggle
  Downloading kaggle-1.5.12.tar.gz (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.0/59.0 kB 1.3 MB/s eta 0:00:00ta 0:00:01
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six>=1.10 in /opt/conda/lib/python3.7/site-packages (from kaggle) (1.14.0)
Requirement already satisfied: certifi in /opt/conda/lib/python3.7/site-packages (from kaggle) (2022.9.24)
Requirement already satisfied: python-dateutil in /opt/conda/lib/python3.7/site-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-packages (from kaggle) (2.28.1)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.7/site-packages (from kaggle) (4.42.1)
Collecting python-slugify
  Downloading python_slugify-8.0.0-py2.py3-none-any.whl (9.5 kB)
Requirement already satisfied: urllib3 in /opt/conda/lib/python3.7/site-packages (from kaggle) (1.26.13)
Collecting text-unidecode>=1.3
  Downloading text_unidecode-1.3-py2.py3-none-any.whl (78 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.2/78.2 kB 1.6 MB/s eta 0:00:00ta 0:00:01
Requirement already satisfied: charset-normalizer<3,>=2 in /opt/conda/lib/python3.7/site-packages (from requests->kaggle) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.7/site-packages (from requests->kaggle) (2.8)
Building wheels for collected packages: kaggle
  Building wheel for kaggle (setup.py) ... done
  Created wheel for kaggle: filename=kaggle-1.5.12-py3-none-any.whl size=73052 sha256=84318f0291db295d9ebd2181d4c19eb9b4b25bd213fb8268dc
c13bf2dcfba399
  Stored in directory: /root/.cache/pip/wheels/11/ec/8f/80c32ff2501f7b1a76f4df651a0242314d229a5d3e5130bd01
Successfully built kaggle
Installing collected packages: text-unidecode, python-slugify, kaggle
Successfully installed kaggle-1.5.12 python-slugify-8.0.0 text-unidecode-1.3
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It i
s recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip available: 22.3.1 -> 23.0
[notice] To update, run: pip install --upgrade pip
```

In [5]:
```
!mkdir -p /root/.kaggle
!touch /root/.kaggle/kaggle.json
!chmod 600 /root/.kaggle/kaggle.json
```

In [6]:
```
# Fill in your user name and key from creating the kaggle account and API token file
import json
kaggle_username = "user_name"
kaggle_key = "key"

# Save API token the kaggle.json file
with open("/root/.kaggle/kaggle.json", "w") as f:
    f.write(json.dumps({"username": kaggle_username, "key": kaggle_key}))
```

In [8]:
```
!kaggle datasets download -d shadman0786/asl-alphabet-cnn
```

```
Downloading asl-alphabet-cnn.zip to /root
100%|█████████████████████████████████████▉| 1.02G/1.02G [00:04<00:00, 226MB/s]
100%|██████████████████████████████████████| 1.02G/1.02G [00:10<00:00, 100MB/s]
```

In [9]:
```
# unzipping dogs_vs_cats.zip file using quiet
!unzip -q asl-alphabet-cnn.zip # -q for quiet
```

In [ ]:

In [10…
```
# uploading data in AWS S3
import time
tic = time.clock()
prefix ="asl_alphabet"
print("Starting to upload asl_alphabet")

inputs = sagemaker_session.upload_data(path="asl_alphabet", bucket=bucket, key_prefix=prefix)
print(f"Input path ( S3 file path ): {inputs}")
toc = time.clock()
print(f'time consumed: {toc - tic}')
```

```
Starting to upload asl_alphabet
Input path ( S3 file path ): s3://sagemaker-us-east-1-910293992207/asl_alphabet
```

In [9]:
```
inputs = "s3://sagemaker-us-east-1-910293992207/asl_alphabet"
print(f"Input path ( S3 file path ): {inputs}")
```

```
Input path ( S3 file path ): s3://sagemaker-us-east-1-910293992207/asl_alphabet
```

In [11…
```
path = "./asl_alphabet/train"
path
```

Out[11]:
```
'./asl_alphabet/train'
```

Data Visualization

In [24…

```
from PIL import Image
import io
import os
import numpy as np

path = ["./asl_alphabet/train/A/A1000.jpg"
        ,"./asl_alphabet/train/B/B1000.jpg"
       ,"./asl_alphabet/train/C/C1000.jpg"
       ,"./asl_alphabet/train/D/D1000.jpg"
       ,"./asl_alphabet/train/E/E1000.jpg"
       ,"./asl_alphabet/train/F/F1000.jpg"
       ,"./asl_alphabet/train/G/G1000.jpg"
       ,"./asl_alphabet/train/H/H1000.jpg"
       ,"./asl_alphabet/train/I/I1000.jpg"
       ,"./asl_alphabet/train/J/J1000.jpg"
       ,"./asl_alphabet/train/K/K1000.jpg"
       ,"./asl_alphabet/train/L/L1000.jpg"
       ,"./asl_alphabet/train/M/M1000.jpg"
       ,"./asl_alphabet/train/N/N1000.jpg"
       ,"./asl_alphabet/train/O/O1000.jpg"
       ,"./asl_alphabet/train/P/P1000.jpg"
       ,"./asl_alphabet/train/Q/Q1000.jpg"
       ,"./asl_alphabet/train/R/R1000.jpg"
       ,"./asl_alphabet/train/S/S1000.jpg"
       ,"./asl_alphabet/train/T/T1000.jpg"
       ,"./asl_alphabet/train/U/U1000.jpg"
       ,"./asl_alphabet/train/V/V1000.jpg"
       ,"./asl_alphabet/train/W/W1000.jpg"
       ,"./asl_alphabet/train/X/X1000.jpg"
       ,"./asl_alphabet/train/Y/Y1000.jpg"
       ,"./asl_alphabet/train/Z/Z1000.jpg"]
#address = ["/A601", "/A602"]
for a in path:
    with open(a , "rb") as f:
        payload = f.read()
        print(f"Category shown below is : {a[23:24]}")
        display(Image.open(io.BytesIO(payload)))
```

```
Category shown below is : A
```

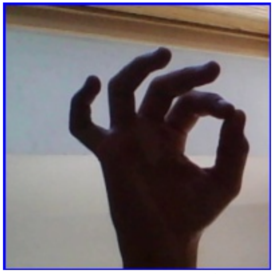
```
Category shown below is : B
```

```
Category shown below is : C
```

```
Category shown below is : D
```

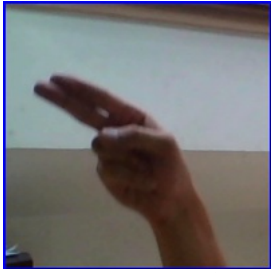
```
Category shown below is : E
```

```
Category shown below is : F
```

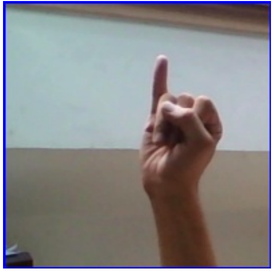


```
Category shown below is : G
```

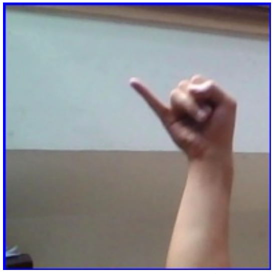
```
Category shown below is : H
```



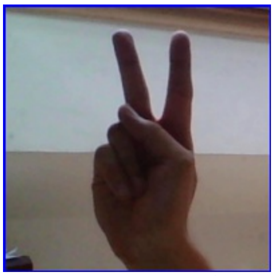
```
Category shown below is : I
```



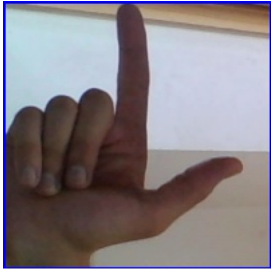
```
Category shown below is : J
```



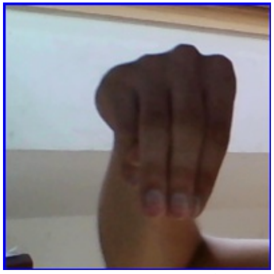
```
Category shown below is : K
```



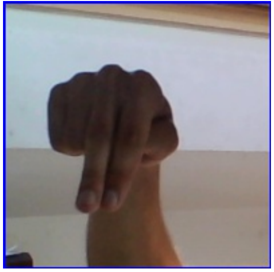
```
Category shown below is : L
```



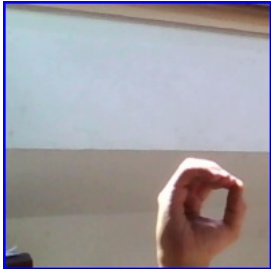
```
Category shown below is : M
```



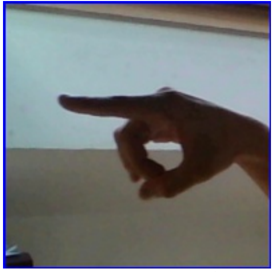
```
Category shown below is : N
```



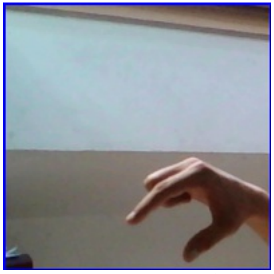
```
Category shown below is : O
```



```
Category shown below is : P
```



```
Category shown below is : Q
```



```
Category shown below is : R
```

```
Category shown below is : S
```

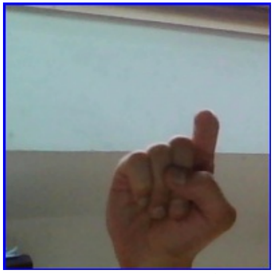
```
Category shown below is : T
```

```
Category shown below is : U
```

```
Category shown below is : V
```

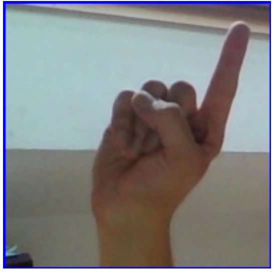
```
Category shown below is : W
```

```
Category shown below is : X
```



```
Category shown below is : Y
```

```
Category shown below is : Z
```



Hyperparameter Tuning

The ResNet50 model with a two Fully connected Linear NN layer's is used for this image classification problem. ResNet-50 is 50 layers deep and is trained on a million images of 1000 categories from the ImageNet database. Furthermore the model has a lot of trainable parameters, which indicates a deep architecture that makes it better for image recognition The optimizer that we will be using for this model is AdamW (For more info refer : https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html (https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html)) Hence, the hyperparameters selected for tuning were: Learning rate - default(x) is 0.001 , so we have selected 0.01x to 100x range for the learing rate eps - defaut is 1e-08 , which is acceptable in most cases so we have selected a range of 1e-09 to 1e-08 Weight decay - default(x) is 0.01 , so we have selected 0.1x to 10x range for the weight decay Batch size -- selected only two values [64, 128]

In [5]:

```
#Importing all the required modules fomr tuner
from sagemaker.tuner import (
    CategoricalParameter,
    ContinuousParameter,
    HyperparameterTuner
)

# We wil be using AdamW as an optimizer which uses a different( more correct or better) way to calulate the weight decay rel
ated computations
# So we will be using weight_decay and eps hyperparamter tuning as well , along with the lerning rate and batchsize params
hyperparameter_ranges = {
    "lr": ContinuousParameter(0.0001, 0.1),
    "eps": ContinuousParameter(1e-9, 1e-8),
    "weight_decay": ContinuousParameter(1e-3, 1e-1),
    "batch_size": CategoricalParameter([ 64, 128]),
}
objective_metric_name = "average test loss"
objective_type = "Minimize"
metric_definitions = [{"Name": "average test loss", "Regex": "Test set: Average loss: ([0-9\\.]+)"}]
```

In [6]:

```
from  sagemaker.pytorch import PyTorch

estimator = PyTorch(
    entry_point = "hpo.py",
    base_job_name = "asl-alphabet-hpo",
    role = role,
    instance_count = 1,
    instance_type = "ml.g4dn.xlarge",
    py_version = "py36",
    framework_version = "1.8"
)

tuner = HyperparameterTuner(
    estimator,
    objective_metric_name,
    hyperparameter_ranges,
    metric_definitions,
    max_jobs=4,
    max_parallel_jobs=1,
    objective_type=objective_type,
    early_stopping_type="Auto"
)
```

In [ ]:

```
# TODO: Fit your HP Tuner
tuner.fit({"training": inputs }, wait=True)
```

```
No finished training job found associated with this estimator. Please make sure this estimator is only used for building workflow config
No finished training job found associated with this estimator. Please make sure this estimator is only used for building workflow config
...............................................................................................................................
```

In [8]:

```
# Get the best estimators and the best HPs

best_estimator = tuner.best_estimator()

#Get the hyperparameters of the best trained model
best_estimator.hyperparameters()
```

```
2023-02-08 11:32:06 Starting - Found matching resource for reuse
2023-02-08 11:32:06 Downloading - Downloading input data
2023-02-08 11:32:06 Training - Training image download completed. Training in progress.
2023-02-08 11:32:06 Uploading - Uploading generated training model
2023-02-08 11:32:06 Completed - Resource reused by training job: pytorch-training-230208-1046-003-f9ccea2c
```

Out[8]:

```
{'_tuning_objective_metric': '"average test loss"',
 'batch_size': '"64"',
 'eps': '8.894659223977433e-09',
 'lr': '0.0009034645151607949',
 'sagemaker_container_log_level': '20',
 'sagemaker_estimator_class_name': '"PyTorch"',
 'sagemaker_estimator_module': '"sagemaker.pytorch.estimator"',
 'sagemaker_job_name': '"dog-cat-classification-hpo-2023-02-08-10-46-41-371"',
 'sagemaker_program': '"hpo.py"',
 'sagemaker_region': '"us-east-1"',
 'sagemaker_submit_directory': '"s3://sagemaker-us-east-1-910293992207/dog-cat-classification-hpo-2023-02-08-10-46-41-371/source/sourced
ir.tar.gz"',
 'weight_decay': '0.0029532014437717905'}
```

In [9]:

```
# Below are the hyperparameters markdown, that can be used instead of re-running the entire hypertuning job
```

In [12…

```
{'batch_size': 64, 'eps': '8.894659223977433e-09', 'lr': '0.0009034645151607949', 'weight_decay': '0.0029532014437717905'}
```

Out[12]:

```
{'batch_size': 64,
 'eps': '8.894659223977433e-09',
 'lr': '0.0009034645151607949',
 'weight_decay': '0.0029532014437717905'}
```

In [13…

```
best_hyperparameters={'batch_size': int(best_estimator.hyperparameters()['batch_size'].replace('"', "")),
                      'eps': best_estimator.hyperparameters()['eps'],
                      'lr': best_estimator.hyperparameters()['lr'],
                      'weight_decay': best_estimator.hyperparameters()['weight_decay'],}
print(f"Best Hyperparamters post Hyperparameter fine tuning are : \n {best_hyperparameters}")
```

```
Best Hyperparamters post Hyperparameter fine tuning are :
 {'batch_size': 64, 'eps': '8.894659223977433e-09', 'lr': '0.0009034645151607949', 'weight_decay': '0.0029532014437717905'}
```

Model Profiling and Debugging

Note: You will need to use the `train_model.py` script to perform model profiling and debugging.

In [10…

```
# Setting up debugger and profiler rules and configs
from  sagemaker.pytorch import PyTorch
from sagemaker.debugger import (
    Rule,
    rule_configs,
    ProfilerRule,
    DebuggerHookConfig,
    CollectionConfig,
    ProfilerConfig,
    FrameworkProfile
)


rules = [
    Rule.sagemaker(rule_configs.vanishing_gradient()),
    Rule.sagemaker(rule_configs.overfit()),
    Rule.sagemaker(rule_configs.overtraining()),
    Rule.sagemaker(rule_configs.poor_weight_initialization()),
    ProfilerRule.sagemaker(rule_configs.ProfilerReport()),
]

profiler_config = ProfilerConfig(
    system_monitor_interval_millis=500, framework_profile_params=FrameworkProfile(num_steps=10)
)

collection_configs=[CollectionConfig(name="CrossEntropyLoss_output_0",parameters={
    "include_regex": "CrossEntropyLoss_output_0", "train.save_interval": "10","eval.save_interval": "1"})]

debugger_config=DebuggerHookConfig( collection_configs=collection_configs )
```

In [11…

```
# Create and fit an estimator
estimator = PyTorch(
    entry_point="train_model.py",
    instance_count=1,
    instance_type="ml.g4dn.xlarge",
    role=role,
    framework_version="1.6", #using 1.6 as it has support for smdebug lib , https://github.com/awslabs/sagemaker-debugger#debugger-supported-frameworks
    py_version="py36",
    hyperparameters=best_hyperparameters,
    profiler_config=profiler_config, # include the profiler hook
    debugger_hook_config=debugger_config, # include the debugger hook
    rules=rules
)

estimator.fit({'train' : inputs },wait=True)
```

```
2023-02-10 05:01:23 Starting - Starting the training job...
2023-02-10 05:01:38 Starting - Preparing the instances for trainingVanishingGradient: InProgress
Overfit: InProgress
Overtraining: InProgress
PoorWeightInitialization: InProgress
ProfilerReport: InProgress
......
2023-02-10 05:02:51 Downloading - Downloading input data.........
2023-02-10 05:04:25 Training - Downloading the training image......
2023-02-10 05:05:19 Training - Training image download completed. Training in progress...bash: cannot set terminal process group (-1): I
nappropriate ioctl for device
bash: no job control in this shell
2023-02-10 05:05:30,156 sagemaker-training-toolkit INFO     Imported framework sagemaker_pytorch_container.training
2023-02-10 05:05:30,184 sagemaker_pytorch_container.training INFO     Block until all host DNS lookups succeed.
2023-02-10 05:05:30,187 sagemaker_pytorch_container.training INFO     Invoking user training script.
2023-02-10 05:05:30,426 sagemaker-training-toolkit INFO     Invoking user script
Training Env:
{
    "additional_framework_parameters": {},
    "channel_input_dirs": {
        "train": "/opt/ml/input/data/train"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_pytorch_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {
        "batch_size": 64,
        "eps": "8.894659223977433e-09",
        "lr": "0.0009034645151607949",
        "weight_decay": "0.0029532014437717905"
    },
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "train": {
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
    "job_name": "pytorch-training-2023-02-10-05-01-23-323",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/source/sourcedir.tar.gz",
    "module_name": "train_model",
    "network_interface_name": "eth0",
    "num_cpus": 4,
    "num_gpus": 1,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "current_instance_type": "ml.g4dn.xlarge",
        "current_group_name": "homogeneousCluster",
        "hosts": [
            "algo-1"
        ],
        "instance_groups": [
            {
                "instance_group_name": "homogeneousCluster",
                "instance_type": "ml.g4dn.xlarge",
                "hosts": [
                    "algo-1"
                ]
            }
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "train_model.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"batch_size":64,"eps":"8.894659223977433e-09","lr":"0.0009034645151607949","weight_decay":"0.0029532014437717905"}
SM_USER_ENTRY_POINT=train_model.py
SM_FRAMEWORK_PARAMS={}
SM_RESOURCE_CONFIG={"current_group_name":"homogeneousCluster","current_host":"algo-1","current_instance_type":"ml.g4dn.xlarge","hosts":
["algo-1"],"instance_groups":[{"hosts":["algo-1"],"instance_group_name":"homogeneousCluster","instance_type":"ml.g4dn.xlarge"}],"network
_interface_name":"eth0"}
SM_INPUT_DATA_CONFIG={"train":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["train"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=train_model
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_pytorch_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=4
SM_NUM_GPUS=1
```

```
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/source/sourcedir.tar.gz
SM_TRAINING_ENV={"additional_framework_parameters":{},"channel_input_dirs":{"train":"/opt/ml/input/data/train"},"current_host":"algo-
1","framework_module":"sagemaker_pytorch_container.training:main","hosts":["algo-1"],"hyperparameters":{"batch_size":64,"eps":"8.8946592
23977433e-09","lr":"0.0009034645151607949","weight_decay":"0.0029532014437717905"},"input_config_dir":"/opt/ml/input/config","input_data
_config":{"train":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}},"input_dir":"/opt/ml/i
nput","is_master":true,"job_name":"pytorch-training-2023-02-10-05-01-23-323","log_level":20,"master_hostname":"algo-1","model_dir":"/op
t/ml/model","module_dir":"s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/source/sourcedir.tar.gz","modul
e_name":"train_model","network_interface_name":"eth0","num_cpus":4,"num_gpus":1,"output_data_dir":"/opt/ml/output/data","output_dir":"/o
pt/ml/output","output_intermediate_dir":"/opt/ml/output/intermediate","resource_config":{"current_group_name":"homogeneousCluster","curr
ent_host":"algo-1","current_instance_type":"ml.g4dn.xlarge","hosts":["algo-1"],"instance_groups":[{"hosts":["algo-1"],"instance_group_na
me":"homogeneousCluster","instance_type":"ml.g4dn.xlarge"}],"network_interface_name":"eth0"},"user_entry_point":"train_model.py"}
SM_USER_ARGS=["--batch_size","64","--eps","8.894659223977433e-09","--lr","0.0009034645151607949","--weight_decay","0.002953201443771790
5"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_BATCH_SIZE=64
SM_HP_EPS=8.894659223977433e-09
SM_HP_LR=0.0009034645151607949
SM_HP_WEIGHT_DECAY=0.0029532014437717905
PYTHONPATH=/opt/ml/code:/opt/conda/bin:/opt/conda/lib/python36.zip:/opt/conda/lib/python3.6:/opt/conda/lib/python3.6/lib-dynload:/opt/co
nda/lib/python3.6/site-packages
Invoking script with the following command:
/opt/conda/bin/python3.6 train_model.py --batch_size 64 --eps 8.894659223977433e-09 --lr 0.0009034645151607949 --weight_decay 0.00295320
14437717905
[2023-02-10 05:05:31.153 algo-1:27 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2023-02-10 05:05:31.347 algo-1:27 INFO profiler_config_parser.py:102] Using config at /opt/ml/input/config/profilerconfig.json.
Running on Device cuda:0
Hyperparameters : LR: 0.0009034645151607949,  Eps: 8.894659223977433e-09, Weight-decay: 0.0029532014437717905, Batch Size: 64, Epoch: 2
Data Dir Path: /opt/ml/input/data/train
Model Dir  Path: /opt/ml/model
Output Dir  Path: /opt/ml/output/data
[2023-02-10 05:05:35.193 algo-1:27 INFO json_config.py:91] Creating hook from json_config at /opt/ml/input/config/debughookconfig.json.
[2023-02-10 05:05:35.195 algo-1:27 INFO hook.py:199] tensorboard_dir has not been set for the hook. SMDebug will not be exporting tensor
board summaries.
[2023-02-10 05:05:35.196 algo-1:27 INFO hook.py:253] Saving to /opt/ml/output/tensors
[2023-02-10 05:05:35.196 algo-1:27 INFO state_store.py:77] The checkpoint config file /opt/ml/input/config/checkpointconfig.json does no
t exist.
[2023-02-10 05:05:35.224 algo-1:27 INFO hook.py:584] name:fc.0.weight count_params:524288
[2023-02-10 05:05:35.224 algo-1:27 INFO hook.py:584] name:fc.0.bias count_params:256
[2023-02-10 05:05:35.225 algo-1:27 INFO hook.py:584] name:fc.2.weight count_params:34048
[2023-02-10 05:05:35.225 algo-1:27 INFO hook.py:584] name:fc.2.bias count_params:133
[2023-02-10 05:05:35.225 algo-1:27 INFO hook.py:586] Total Trainable Params: 558725
Epoch 1 - Starting Training phase.
Epoch: 1 - Training Model on Complete Training Dataset!
[2023-02-10 05:05:38.981 algo-1:27 INFO hook.py:413] Monitoring the collections: gradients, CrossEntropyLoss_output_0, losses, relu_inpu
t
[2023-02-10 05:05:38.983 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/prestepzero-*-start-1676005531348249.5_train-0-stepstart-1676005538983138.2/python_stats.
[2023-02-10 05:05:38.996 algo-1:27 INFO hook.py:476] Hook is writing from the hook with pid: 27
[2023-02-10 05:05:49.956 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-0-stepstart-1676005538994220.0_train-0-forwardpassend-1676005549956186.8/python_stats.
[2023-02-10 05:05:50.351 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-0-forwardpassend-1676005549959238.0_train-1-stepstart-1676005550350463.0/python_stats.
[2023-02-10 05:05:53.678 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-1-stepstart-1676005550355889.0_train-1-forwardpassend-1676005553678065.0/python_stats.
[2023-02-10 05:05:54.161 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-1-forwardpassend-1676005553680365.5_train-2-stepstart-1676005554160409.5/python_stats.
[2023-02-10 05:05:57.659 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-2-stepstart-1676005554163427.5_train-2-forwardpassend-1676005557658744.2/python_stats.
[2023-02-10 05:05:58.099 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-2-forwardpassend-1676005557660861.0_train-3-stepstart-1676005558099099.5/python_stats.
[2023-02-10 05:06:01.247 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-3-stepstart-1676005558102197.2_train-3-forwardpassend-1676005561246245.2/python_stats.
[2023-02-10 05:06:01.710 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-3-forwardpassend-1676005561251337.8_train-4-stepstart-1676005561708644.2/python_stats.
[2023-02-10 05:06:04.916 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-4-stepstart-1676005561717425.0_train-4-forwardpassend-1676005564916602.5/python_stats.
[2023-02-10 05:06:05.350 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-4-forwardpassend-1676005564918364.0_train-5-stepstart-1676005565349659.0/python_stats.
[2023-02-10 05:06:08.437 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-5-stepstart-1676005565353806.2_train-5-forwardpassend-1676005568436846.2/python_stats.
[2023-02-10 05:06:08.747 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-5-forwardpassend-1676005568438501.5_train-6-stepstart-1676005568747124.2/python_stats.
[2023-02-10 05:06:11.481 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-6-stepstart-1676005568749797.5_train-6-forwardpassend-1676005571480755.5/python_stats.
[2023-02-10 05:06:11.792 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-6-forwardpassend-1676005571482341.0_train-7-stepstart-1676005571792309.5/python_stats.
[2023-02-10 05:06:14.475 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-7-stepstart-1676005571795434.2_train-7-forwardpassend-1676005574474824.5/python_stats.
[2023-02-10 05:06:14.789 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-7-forwardpassend-1676005574476509.0_train-8-stepstart-1676005574789312.2/python_stats.
[2023-02-10 05:06:17.499 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-8-stepstart-1676005574792091.2_train-8-forwardpassend-1676005577499157.0/python_stats.
[2023-02-10 05:06:17.810 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-8-forwardpassend-1676005577500818.0_train-9-stepstart-1676005577810100.0/python_stats.
[2023-02-10 05:06:20.547 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-9-stepstart-1676005577812992.0_train-9-forwardpassend-1676005580547446.8/python_stats.
[2023-02-10 05:06:20.866 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-9-forwardpassend-1676005580549297.0_train-10-stepstart-1676005580865954.5/python_stats.
Train set:  [8000/69600 (11%)]#011 Loss: 3.19#011Accuracy: 1437/8000 (17.96%)
Train set:  [16000/69600 (23%)]#011 Loss: 2.52#011Accuracy: 4451/16000 (27.82%)
Train set:  [24000/69600 (34%)]#011 Loss: 2.60#011Accuracy: 8011/24000 (33.38%)
Train set:  [32000/69600 (46%)]#011 Loss: 2.73#011Accuracy: 11815/32000 (36.92%)
Train set:  [40000/69600 (57%)]#011 Loss: 1.76#011Accuracy: 15626/40000 (39.06%)
Train set:  [48000/69600 (69%)]#011 Loss: 2.01#011Accuracy: 19583/48000 (40.80%)
Train set:  [56000/69600 (80%)]#011 Loss: 2.47#011Accuracy: 23517/56000 (41.99%)
Train set:  [64000/69600 (92%)]#011 Loss: 1.98#011Accuracy: 27537/64000 (43.03%)
Train set: Average loss: 2.6271, Accuracy: 30458/69600 (44%)
Epoch 1 - Starting Testing phase.
Epoch: 1 - Testing Model on Complete Testing Dataset!
VanishingGradient: InProgress
Overfit: InProgress
Overtraining: InProgress
PoorWeightInitialization: IssuesFound
Test set: Average loss: 2.1956, Accuracy: 9305/17400 (53%)
Epoch 2 - Starting Training phase.
Epoch: 2 - Training Model on Complete Training Dataset!
Train set:  [8000/69600 (11%)]#011 Loss: 2.27#011Accuracy: 4110/8000 (51.38%)
Train set:  [16000/69600 (23%)]#011 Loss: 1.89#011Accuracy: 8312/16000 (51.95%)
Train set:  [24000/69600 (34%)]#011 Loss: 2.42#011Accuracy: 12643/24000 (52.68%)
Train set:  [32000/69600 (46%)]#011 Loss: 2.80#011Accuracy: 16817/32000 (52.55%)
VanishingGradient: InProgress
Overfit: InProgress
Overtraining: IssuesFound
PoorWeightInitialization: IssuesFound
Train set:  [40000/69600 (57%)]#011 Loss: 2.02#011Accuracy: 21035/40000 (52.59%)
Train set:  [48000/69600 (69%)]#011 Loss: 2.53#011Accuracy: 25315/48000 (52.74%)
Train set:  [56000/69600 (80%)]#011 Loss: 2.06#011Accuracy: 29648/56000 (52.94%)
Train set:  [64000/69600 (92%)]#011 Loss: 2.39#011Accuracy: 34039/64000 (53.19%)
Train set: Average loss: 2.1803, Accuracy: 37030/69600 (53%)
Epoch 2 - Starting Testing phase.
Epoch: 2 - Testing Model on Complete Testing Dataset!
Test set: Average loss: 2.0959, Accuracy: 9534/17400 (55%)
Starting to Save the Model
Completed Saving the Model
INFO:__main__:Running on Device cuda:0
INFO:__main__:Hyperparameters : LR: 0.0009034645151607949,  Eps: 8.894659223977433e-09, Weight-decay: 0.0029532014437717905, Batch Size:
64, Epoch: 2
INFO:__main__:Data Dir Path: /opt/ml/input/data/train
INFO:__main__:Model Dir  Path: /opt/ml/model
INFO:__main__:Output Dir  Path: /opt/ml/output/data
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /root/.cache/torch/hub/checkpoints/resnet50-19c8e357.pth
#015  0%|          | 0.00/97.8M [00:00<?, ?B/s]#015 10%|█         | 9.95M/97.8M [00:00<00:00, 104MB/s]#015 21%|██        | 20.1M/97.8M
[00:00<00:00, 105MB/s]#015 31%|███       | 30.3M/97.8M [00:00<00:00, 105MB/s]#015 40%|████      | 39.1M/97.8M [00:00<00:00, 101MB/s]#015
49%|████▉     | 47.6M/97.8M [00:00<00:00, 97.3MB/s]#015 57%|█████▋    | 56.1M/97.8M [00:00<00:00, 94.8MB/s]#015 66%|██████▌   | 64.7M/
97.8M [00:00<00:00, 93.1MB/s]#015 75%|███████▍  | 73.0M/97.8M [00:00<00:00, 91.4MB/s]#015 83%|████████▎ | 81.6M/97.8M [00:00<00:00, 9
0.9MB/s]#015 92%|█████████▏| 90.2M/97.8M [00:01<00:00, 90.7MB/s]#015100%|██████████| 97.8M/97.8M [00:01<00:00, 93.9MB/s]
/opt/conda/lib/python3.6/site-packages/torch/cuda/__init__.py:125: UserWarning:
Tesla T4 with CUDA capability sm_75 is not compatible with the current PyTorch installation.
The current PyTorch install supports CUDA capabilities sm_35 sm_52 sm_60 sm_61 sm_70 compute_70.
If you want to use the Tesla T4 GPU with PyTorch, please check the instructions at https://pytorch.org/get-started/locally/
  warnings.warn(incompatible_device_warn.format(device_name, capability, " ".join(arch_list), device_name))
INFO:__main__:Epoch 1 - Starting Training phase.
INFO:__main__:Epoch: 1 - Training Model on Complete Training Dataset!
INFO:__main__:
Train set:  [8000/69600 (11%)]#011 Loss: 3.19#011Accuracy: 1437/8000 (17.96%)
INFO:__main__:
Train set:  [16000/69600 (23%)]#011 Loss: 2.52#011Accuracy: 4451/16000 (27.82%)
INFO:__main__:
Train set:  [24000/69600 (34%)]#011 Loss: 2.60#011Accuracy: 8011/24000 (33.38%)
INFO:__main__:
Train set:  [32000/69600 (46%)]#011 Loss: 2.73#011Accuracy: 11815/32000 (36.92%)
INFO:__main__:
Train set:  [40000/69600 (57%)]#011 Loss: 1.76#011Accuracy: 15626/40000 (39.06%)
INFO:__main__:
Train set:  [48000/69600 (69%)]#011 Loss: 2.01#011Accuracy: 19583/48000 (40.80%)
INFO:__main__:
Train set:  [56000/69600 (80%)]#011 Loss: 2.47#011Accuracy: 23517/56000 (41.99%)
INFO:__main__:
Train set:  [64000/69600 (92%)]#011 Loss: 1.98#011Accuracy: 27537/64000 (43.03%)
INFO:__main__:
Train set: Average loss: 2.6271, Accuracy: 30458/69600 (44%)
INFO:__main__:Epoch 1 - Starting Testing phase.
INFO:__main__:Epoch: 1 - Testing Model on Complete Testing Dataset!
INFO:__main__:
Test set: Average loss: 2.1956, Accuracy: 9305/17400 (53%)
INFO:__main__:Epoch 2 - Starting Training phase.
INFO:__main__:Epoch: 2 - Training Model on Complete Training Dataset!
INFO:__main__:
Train set:  [8000/69600 (11%)]#011 Loss: 2.27#011Accuracy: 4110/8000 (51.38%)
INFO:__main__:
Train set:  [16000/69600 (23%)]#011 Loss: 1.89#011Accuracy: 8312/16000 (51.95%)
INFO:__main__:
Train set:  [24000/69600 (34%)]#011 Loss: 2.42#011Accuracy: 12643/24000 (52.68%)
INFO:__main__:
Train set:  [32000/69600 (46%)]#011 Loss: 2.80#011Accuracy: 16817/32000 (52.55%)
INFO:__main__:
Train set:  [40000/69600 (57%)]#011 Loss: 2.02#011Accuracy: 21035/40000 (52.59%)
INFO:__main__:
Train set:  [48000/69600 (69%)]#011 Loss: 2.53#011Accuracy: 25315/48000 (52.74%)
INFO:__main__:
Train set:  [56000/69600 (80%)]#011 Loss: 2.06#011Accuracy: 29648/56000 (52.94%)
INFO:__main__:
Train set:  [64000/69600 (92%)]#011 Loss: 2.39#011Accuracy: 34039/64000 (53.19%)
INFO:__main__:
Train set: Average loss: 2.1803, Accuracy: 37030/69600 (53%)
INFO:__main__:Epoch 2 - Starting Testing phase.
2023-02-10 05:25:07,380 sagemaker-training-toolkit INFO     Reporting training SUCCESS
INFO:__main__:Epoch: 2 - Testing Model on Complete Testing Dataset!
INFO:__main__:
Test set: Average loss: 2.0959, Accuracy: 9534/17400 (55%)
```

```
INFO:__main__:Starting to Save the Model
INFO:__main__:Completed Saving the Model

2023-02-10 05:25:36 Uploading - Uploading generated training model
2023-02-10 05:25:36 Completed - Training job completed
ProfilerReport: IssuesFound
Training seconds: 1373
Billable seconds: 1373
```

In [12…

```
#fetching jobname , client and description to be used for plotting.
job_name = estimator.latest_training_job.name
client = estimator.sagemaker_session.sagemaker_client
description = client.describe_training_job(TrainingJobName=estimator.latest_training_job.name)
```

In [13…

```
print(f"Jobname: {job_name}")
print(f"Client: {client}")
print(f"Description: {description}")
```

```
Jobname: pytorch-training-2023-02-10-05-01-23-323
Client: <botocore.client.SageMaker object at 0x7fe3998f0090>
Description: {'TrainingJobName': 'pytorch-training-2023-02-10-05-01-23-323', 'TrainingJobArn': 'arn:aws:sagemaker:us-east-1:91029399220
7:training-job/pytorch-training-2023-02-10-05-01-23-323', 'ModelArtifacts': {'S3ModelArtifacts': 's3://sagemaker-us-east-1-910293992207/
pytorch-training-2023-02-10-05-01-23-323/output/model.tar.gz'}, 'TrainingJobStatus': 'Completed', 'SecondaryStatus': 'Completed', 'Hyper
Parameters': {'batch_size': '64', 'eps': '"8.894659223977433e-09"', 'lr': '"0.0009034645151607949"', 'sagemaker_container_log_level': '2
0', 'sagemaker_job_name': '"pytorch-training-2023-02-10-05-01-23-323"', 'sagemaker_program': '"train_model.py"', 'sagemaker_region': '"u
s-east-1"', 'sagemaker_submit_directory': '"s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/source/source
dir.tar.gz"', 'weight_decay': '"0.0029532014437717905"'}, 'AlgorithmSpecification': {'TrainingImage': '763104351884.dkr.ecr.us-east-1.am
azonaws.com/pytorch-training:1.6-gpu-py36', 'TrainingInputMode': 'File', 'EnableSageMakerMetricsTimeSeries': True}, 'RoleArn': 'arn:aws:
iam::910293992207:role/service-role/AmazonSageMaker-ExecutionRole-20230207T185160', 'InputDataConfig': [{'ChannelName': 'train', 'DataSo
urce': {'S3DataSource': {'S3DataType': 'S3Prefix', 'S3Uri': 's3://sagemaker-us-east-1-910293992207/asl_alphabet', 'S3DataDistributionTyp
e': 'FullyReplicated'}}, 'CompressionType': 'None', 'RecordWrapperType': 'None'}], 'OutputDataConfig': {'KmsKeyId': '', 'S3OutputPath':
's3://sagemaker-us-east-1-910293992207/'}, 'ResourceConfig': {'InstanceType': 'ml.g4dn.xlarge', 'InstanceCount': 1, 'VolumeSizeInGB': 3
0}, 'StoppingCondition': {'MaxRuntimeInSeconds': 86400}, 'CreationTime': datetime.datetime(2023, 2, 10, 5, 1, 23, 981000, tzinfo=tzlocal
()), 'TrainingStartTime': datetime.datetime(2023, 2, 10, 5, 2, 39, 886000, tzinfo=tzlocal()), 'TrainingEndTime': datetime.datetime(2023,
2, 10, 5, 25, 32, 621000, tzinfo=tzlocal()), 'LastModifiedTime': datetime.datetime(2023, 2, 10, 5, 25, 48, 207000, tzinfo=tzlocal()), 'S
econdaryStatusTransitions': [{'Status': 'Starting', 'StartTime': datetime.datetime(2023, 2, 10, 5, 1, 23, 981000, tzinfo=tzlocal()), 'En
dTime': datetime.datetime(2023, 2, 10, 5, 2, 39, 886000, tzinfo=tzlocal()), 'StatusMessage': 'Preparing the instances for training'},
{'Status': 'Downloading', 'StartTime': datetime.datetime(2023, 2, 10, 5, 2, 39, 886000, tzinfo=tzlocal()), 'EndTime': datetime.datetime
(2023, 2, 10, 5, 4, 25, 706000, tzinfo=tzlocal()), 'StatusMessage': 'Downloading input data'}, {'Status': 'Training', 'StartTime': datet
ime.datetime(2023, 2, 10, 5, 4, 25, 706000, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 10, 5, 25, 12, 194000, tzinfo=tzloc
al()), 'StatusMessage': 'Training image download completed. Training in progress.'}, {'Status': 'Uploading', 'StartTime': datetime.datet
ime(2023, 2, 10, 5, 25, 12, 194000, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 10, 5, 25, 32, 621000, tzinfo=tzlocal()),
'StatusMessage': 'Uploading generated training model'}, {'Status': 'Completed', 'StartTime': datetime.datetime(2023, 2, 10, 5, 25, 32, 6
21000, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 10, 5, 25, 32, 621000, tzinfo=tzlocal()), 'StatusMessage': 'Training job
completed'}], 'EnableNetworkIsolation': False, 'EnableInterContainerTrafficEncryption': False, 'EnableManagedSpotTraining': False, 'Trai
ningTimeInSeconds': 1373, 'BillableTimeInSeconds': 1373, 'DebugHookConfig': {'S3OutputPath': 's3://sagemaker-us-east-1-910293992207/',
'CollectionConfigurations': [{'CollectionName': 'relu_input', 'CollectionParameters': {'include_regex': '.*relu_input', 'save_interval':
'500'}}, {'CollectionName': 'gradients', 'CollectionParameters': {'save_interval': '500'}}, {'CollectionName': 'CrossEntropyLoss_output_
0', 'CollectionParameters': {'eval.save_interval': '1', 'include_regex': 'CrossEntropyLoss_output_0', 'train.save_interval': '10'}}]},
'DebugRuleConfigurations': [{'RuleConfigurationName': 'VanishingGradient', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazona
ws.com/sagemaker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'VanishingGradient'}}, {'RuleConfigur
ationName': 'Overfit', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.com/sagemaker-debugger-rules:latest', 'VolumeSize
InGB': 0, 'RuleParameters': {'rule_to_invoke': 'Overfit'}}, {'RuleConfigurationName': 'Overtraining', 'RuleEvaluatorImage': '50389593136
0.dkr.ecr.us-east-1.amazonaws.com/sagemaker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'Overtrain
ing'}}, {'RuleConfigurationName': 'PoorWeightInitialization', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.com/sagema
ker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'PoorWeightInitialization'}}], 'DebugRuleEvaluatio
nStatuses': [{'RuleConfigurationName': 'VanishingGradient', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:910293992207:processing
-job/pytorch-training-2023-02-1-vanishinggradient-ef063d1c', 'RuleEvaluationStatus': 'NoIssuesFound', 'LastModifiedTime': datetime.datet
ime(2023, 2, 10, 5, 25, 48, 187000, tzinfo=tzlocal())}, {'RuleConfigurationName': 'Overfit', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:
us-east-1:910293992207:processing-job/pytorch-training-2023-02-1-overfit-6e30cbaf', 'RuleEvaluationStatus': 'NoIssuesFound', 'LastModifi
edTime': datetime.datetime(2023, 2, 10, 5, 25, 48, 187000, tzinfo=tzlocal())}, {'RuleConfigurationName': 'Overtraining', 'RuleEvaluation
JobArn': 'arn:aws:sagemaker:us-east-1:910293992207:processing-job/pytorch-training-2023-02-1-overtraining-27e0aa52', 'RuleEvaluationStat
us': 'IssuesFound', 'StatusDetails': 'RuleEvaluationConditionMet: Evaluation of the rule Overtraining at step 1842 resulted in the condi
tion being met\n', 'LastModifiedTime': datetime.datetime(2023, 2, 10, 5, 25, 48, 187000, tzinfo=tzlocal())}, {'RuleConfigurationName':
'PoorWeightInitialization', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:910293992207:processing-job/pytorch-training-2023-02-1-
poorweightinitialization-d524d4a4', 'RuleEvaluationStatus': 'IssuesFound', 'StatusDetails': 'RuleEvaluationConditionMet: Evaluation of t
he rule PoorWeightInitialization at step 0 resulted in the condition being met\n', 'LastModifiedTime': datetime.datetime(2023, 2, 10, 5,
25, 48, 187000, tzinfo=tzlocal())}], 'ProfilerConfig': {'S3OutputPath': 's3://sagemaker-us-east-1-910293992207/', 'ProfilingIntervalInMi
lliseconds': 500, 'ProfilingParameters': {'DataloaderProfilingConfig': '{"StartStep": 0, "NumSteps": 10, "MetricsRegex": ".*", }', 'Deta
iledProfilingConfig': '{"StartStep": 0, "NumSteps": 10, }', 'FileOpenFailThreshold': '50', 'HorovodProfilingConfig': '{"StartStep": 0,
"NumSteps": 10, }', 'LocalPath': '/opt/ml/output/profiler', 'PythonProfilingConfig': '{"StartStep": 0, "NumSteps": 10, "ProfilerName":
"cprofile", "cProfileTimer": "total_time", }', 'RotateFileCloseIntervalInSeconds': '60', 'RotateMaxFileSizeInBytes': '10485760', 'SMData
ParallelProfilingConfig': '{"StartStep": 0, "NumSteps": 10, }'}, 'DisableProfiler': False}, 'ProfilerRuleConfigurations': [{'RuleConfigu
rationName': 'ProfilerReport', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.com/sagemaker-debugger-rules:latest', 'Vo
lumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'ProfilerReport'}}], 'ProfilerRuleEvaluationStatuses': [{'RuleConfigurationName':
'ProfilerReport', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:910293992207:processing-job/pytorch-training-2023-02-1-profilerre
port-9c3460e4', 'RuleEvaluationStatus': 'IssuesFound', 'StatusDetails': 'RuleEvaluationConditionMet: Evaluation of the rule ProfilerRepo
rt at step 23 resulted in the condition being met\n', 'LastModifiedTime': datetime.datetime(2023, 2, 10, 5, 25, 36, 240000, tzinfo=tzloc
al())}], 'ProfilingStatus': 'Enabled', 'ResponseMetadata': {'RequestId': '19fee3cd-6191-487c-93b6-def1b54d3e5e', 'HTTPStatusCode': 200,
'HTTPHeaders': {'x-amzn-requestid': '19fee3cd-6191-487c-93b6-def1b54d3e5e', 'content-type': 'application/x-amz-json-1.1', 'content-lengt
h': '7130', 'date': 'Fri, 10 Feb 2023 05:27:01 GMT'}, 'RetryAttempts': 0}}
```

In [14…

```
from smdebug.trials import create_trial
from smdebug.core.modes import ModeKeys
#creating a trial
trial = create_trial(estimator.latest_job_debugger_artifacts_path())
```

```
[2023-02-10 05:27:14.742 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:17 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: N
one
[2023-02-10 05:27:14.755 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:17 INFO s3_trial.py:42] Loading trial debug-output at
path s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/debug-output
[2023-02-10 05:27:15.340 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:17 WARNING s3handler.py:183] Encountered the exceptio
n An error occurred while reading from response stream: ('Connection broken: IncompleteRead(0 bytes read, 4784 more expected)', Incomple
teRead(0 bytes read, 4784 more expected)) while reading s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/d
ebug-output/index/000000000/000000000960_worker_0.json . Will retry now
```

In [15…

```
trial.tensor_names() #all the tensor names
```

```
[2023-02-10 05:27:24.262 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:17 INFO trial.py:198] Training has ended, will refres
h one final time in 1 sec.
[2023-02-10 05:27:25.286 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:17 INFO trial.py:210] Loaded all steps
```

Out[15]:

```
['CrossEntropyLoss_output_0',
 'gradient/ResNet_fc.0.bias',
 'gradient/ResNet_fc.0.weight',
 'gradient/ResNet_fc.2.bias',
 'gradient/ResNet_fc.2.weight',
 'layer1.0.relu_input_0',
 'layer1.0.relu_input_1',
 'layer1.0.relu_input_2',
 'layer1.1.relu_input_0',
 'layer1.1.relu_input_1',
 'layer1.1.relu_input_2',
 'layer1.2.relu_input_0',
 'layer1.2.relu_input_1',
 'layer1.2.relu_input_2',
 'layer2.0.relu_input_0',
 'layer2.0.relu_input_1',
 'layer2.0.relu_input_2',
 'layer2.1.relu_input_0',
 'layer2.1.relu_input_1',
 'layer2.1.relu_input_2',
 'layer2.2.relu_input_0',
 'layer2.2.relu_input_1',
 'layer2.2.relu_input_2',
 'layer2.3.relu_input_0',
 'layer2.3.relu_input_1',
 'layer2.3.relu_input_2',
 'layer3.0.relu_input_0',
 'layer3.0.relu_input_1',
 'layer3.0.relu_input_2',
 'layer3.1.relu_input_0',
 'layer3.1.relu_input_1',
 'layer3.1.relu_input_2',
 'layer3.2.relu_input_0',
 'layer3.2.relu_input_1',
 'layer3.2.relu_input_2',
 'layer3.3.relu_input_0',
 'layer3.3.relu_input_1',
 'layer3.3.relu_input_2',
 'layer3.4.relu_input_0',
 'layer3.4.relu_input_1',
 'layer3.4.relu_input_2',
 'layer3.5.relu_input_0',
 'layer3.5.relu_input_1',
 'layer3.5.relu_input_2',
 'layer4.0.relu_input_0',
 'layer4.0.relu_input_1',
 'layer4.0.relu_input_2',
 'layer4.1.relu_input_0',
 'layer4.1.relu_input_1',
 'layer4.1.relu_input_2',
 'layer4.2.relu_input_0',
 'layer4.2.relu_input_1',
 'layer4.2.relu_input_2',
 'relu_input_0']
```

In [16…

```
len(trial.tensor("CrossEntropyLoss_output_0").steps(mode=ModeKeys.TRAIN))
```

Out[16]:

```
218
```

In [17…

```
len(trial.tensor("CrossEntropyLoss_output_0").steps(mode=ModeKeys.EVAL))
```

Out[17]:

```
544
```

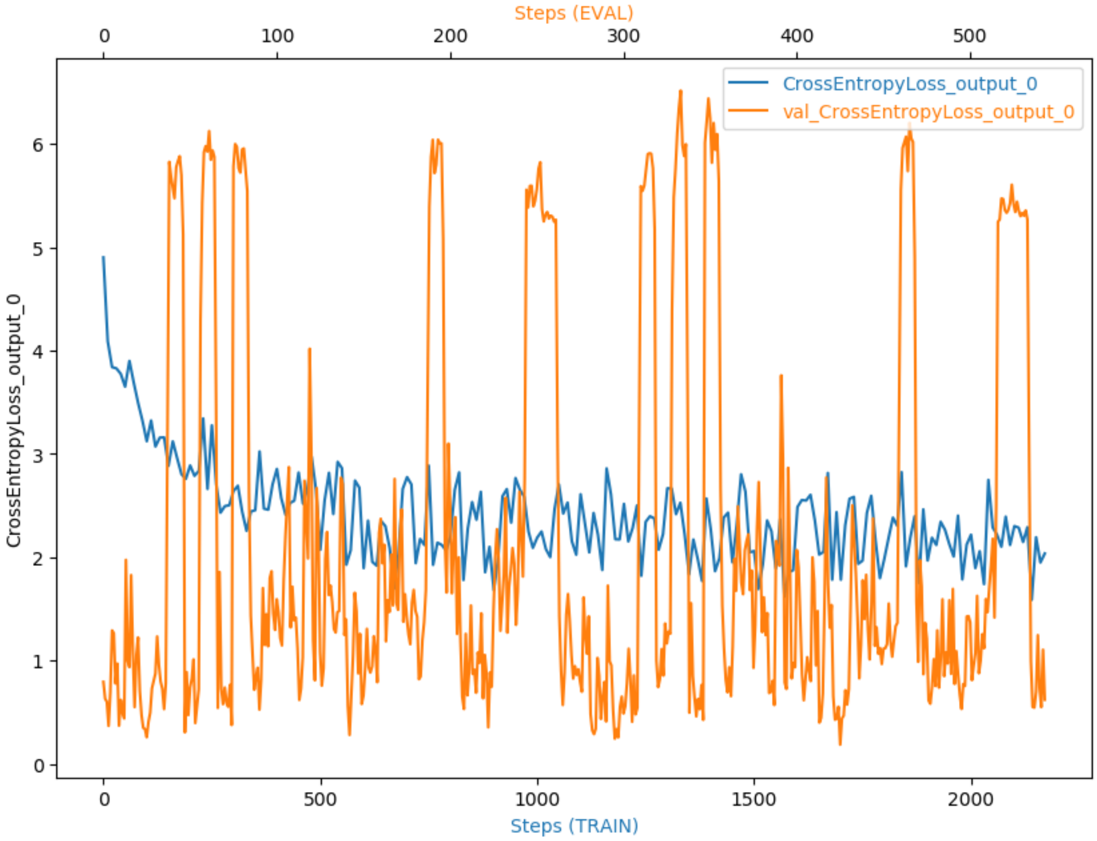
In [18…

```
#Defining some utility functions to be used for plotting tensors
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import host_subplot

#utility function to get data from tensors
def get_data(trial, tname, mode):
    tensor = trial.tensor(tname)
    steps = tensor.steps(mode=mode)
    vals = []
    for s in steps:
        vals.append(tensor.value(s, mode=mode))
    return steps, vals

#plot tensor utility functions for plotting tensors
def plot_tensor(trial, tensor_name):

    steps_train, vals_train = get_data(trial, tensor_name, mode=ModeKeys.TRAIN)
    print("loaded TRAIN data")
    steps_eval, vals_eval = get_data(trial, tensor_name, mode=ModeKeys.EVAL)
    print("loaded EVAL data")

    fig = plt.figure(figsize=(10, 7))
    host = host_subplot(111)

    par = host.twiny()

    host.set_xlabel("Steps (TRAIN)")
    par.set_xlabel("Steps (EVAL)")
    host.set_ylabel(tensor_name)

    (p1,) = host.plot(steps_train, vals_train, label=tensor_name)
    print("Completed TRAIN plot")
    (p2,) = par.plot(steps_eval, vals_eval, label="val_" + tensor_name)
    print("Completed EVAL plot")
    leg = plt.legend()

    host.xaxis.get_label().set_color(p1.get_color())
    leg.texts[0].set_color(p1.get_color())

    par.xaxis.get_label().set_color(p2.get_color())
    leg.texts[1].set_color(p2.get_color())

    plt.ylabel(tensor_name)
    plt.show()
```

In [19…

```
#plotting the tensor
plot_tensor(trial, "CrossEntropyLoss_output_0")
```

```
loaded TRAIN data
loaded EVAL data
Completed TRAIN plot
Completed EVAL plot
```



In [20…

```
# TODO: Display the profiler output
rule_output_path = estimator.output_path + estimator.latest_training_job.job_name + "/rule-output"
print(f"Profiler report location: {rule_output_path}")
```

```
Profiler report location: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output
```

This line of code is a shell command that is listing the contents of a directory on Amazon S3, a cloud storage service. The command has the following parts: !: This is a shell escape character in Jupyter notebooks or IPython, which allows you to run shell commands from within the notebook. aws s3 ls: This is the AWS command line interface (CLI) command to list the contents of an S3 bucket. {rule_output_path}: This is a variable that contains the path to the directory in the S3 bucket you want to list the contents of. --recursive: This is an option to the aws s3 ls command that tells the CLI to list the contents of the directory and all its subdirectories recursively. So, this line of code is running the aws s3 ls command to list the contents of the specified directory in an S3 bucket, including all its subdirectories. In the context of the above shell command, "recursively" means to list the contents of the specified directory, and all its subdirectories, in a tree-like structure. This means that the contents of the specified directory will be displayed, as well as the contents of any subdirectories within that directory, and so on, until the entire contents of the directory tree have been displayed.

In [21…

```
! aws s3 ls {rule_output_path} --recursive
```

```
2023-02-10 05:25:31     418831 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-report.html
2023-02-10 05:25:30     274643 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-report.ipynb
2023-02-10 05:25:25        192 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Batc
hSize.json
2023-02-10 05:25:25      15203 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/CPUB
ottleneck.json
2023-02-10 05:25:25        126 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Data
loader.json
2023-02-10 05:25:26        343 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/GPUM
emoryIncrease.json
2023-02-10 05:25:26       2355 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/IOBo
ttleneck.json
2023-02-10 05:25:26        352 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Load
Balancing.json
2023-02-10 05:25:26        352 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/LowG
PUUtilization.json
2023-02-10 05:25:26        233 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/MaxI
nitializationTime.json
2023-02-10 05:25:26       1310 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Over
allFrameworkMetrics.json
2023-02-10 05:25:26        624 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Over
allSystemUsage.json
2023-02-10 05:25:26       2651 pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/profiler-reports/Step
Outlier.json
```

aws s3 cp and aws s3 ls are two different AWS command line interface (CLI) commands for interacting with Amazon S3, a cloud storage service. aws s3 cp is used to copy files from S3 to your local file system or vice versa. It allows you to transfer one or multiple files from one location to another. aws s3 ls is used to list the contents of a directory in an S3 bucket. It allows you to see the files and directories within a specified directory in the bucket. It does not copy the contents to your local file system. So, the main difference between aws s3 cp and aws s3 ls is that aws s3 cp copies files while aws s3 ls lists the contents of a directory.

In [22…

```
! aws s3 cp {rule_output_path} ./ --recursive
```

```
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/Dataloader.json to ProfilerReport/profiler-output/profiler-reports/Dataloader.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/BatchSize.json to ProfilerReport/profiler-output/profiler-reports/BatchSize.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/CPUBottleneck.json to ProfilerReport/profiler-output/profiler-reports/CPUBottleneck.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/MaxInitializationTime.json to ProfilerReport/profiler-output/profiler-reports/MaxInitializationTime.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/GPUMemoryIncrease.json to ProfilerReport/profiler-output/profiler-reports/GPUMemoryIncrease.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/LoadBalancing.json to ProfilerReport/profiler-output/profiler-reports/LoadBalancing.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/LowGPUUtilization.json to ProfilerReport/profiler-output/profiler-reports/LowGPUUtilization.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/OverallFrameworkMetrics.json to ProfilerReport/profiler-output/profiler-reports/OverallFrameworkMetrics.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-report.ipynb to ProfilerReport/profiler-output/profiler-report.ipynb
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/IOBottleneck.json to ProfilerReport/profiler-output/profiler-reports/IOBottleneck.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/OverallSystemUsage.json to ProfilerReport/profiler-output/profiler-reports/OverallSystemUsage.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-reports/StepOutlier.json to ProfilerReport/profiler-output/profiler-reports/StepOutlier.json
download: s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/rule-output/ProfilerReport/profiler-output/prof
iler-report.html to ProfilerReport/profiler-output/profiler-report.html
```

In [23…

```
import os

# get the autogenerated folder name of profiler report
profiler_report_name = [
    rule["RuleConfigurationName"]
    for rule in estimator.latest_training_job.rule_job_summary()
    if "Profiler" in rule["RuleConfigurationName"]
][0]
```

In [24…

```
# Zipping the ProfilerReport inorder to export and upload it later for submission
import shutil
shutil.make_archive("./profiler_report", "zip", "ProfilerReport")
```

Out[24]:

```
'/root/profiler_report.zip'
```

In [61…

```
import IPython

IPython.display.HTML(filename=profiler_report_name + "/profiler-output/profiler-report.html")
```

Out[61]:

SageMaker Debugger Profiling Report

SageMaker Debugger auto generated this report. You can generate similar reports on all supported training jobs. The report provides summary of training jo framework metrics, rules summary, and detailed analysis from each rule. The graphs and tables are interactive.

Legal disclaimer: This report and any recommendations are provided for informational purposes only and are not definitive. You are responsible for making the information.

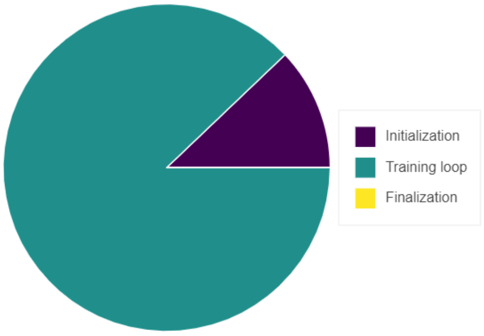
In [4]:

```
# Parameters
processing_job_arn = "arn:aws:sagemaker:us-east-1:910293992207:processing-job/pytorch-training-2023-02-1-profilerreport-9c346
```

Training job summary

The following table gives a summary about the training job. The table includes information about when the training job started and ended, how much time initialization, training loop and finalization took. Your training job started on 02/10/2023 at 05:02:51 and ran for 1328 seconds.

#		Job Statistics
0	Start time	05:02:51 02/10/2023
1	End time	05:24:59 02/10/2023
2	Job duration	1328 seconds
3	Training loop start	05:05:38 02/10/2023
4	Training loop end	05:24:59 02/10/2023
5	Training loop duration	1160 seconds
6	Initialization time	167 seconds
7	Finalization time	0 seconds
8	Initialization	12 %
9	Training loop	87 %
10	Finalization	0 %



System usage statistics

The median total GPU utilization on node algo-1 is 35%. The median total CPU utilization is 40%.

The following table shows statistics of resource utilization per worker (node), such as the total CPU and GPU utilization, and the memory utilization on CPU and GPU. The table also includes the total I/O wait time and the total amount of data sent or received in bytes. The table shows min and max values as well as p99, p90 and p50 percentiles.

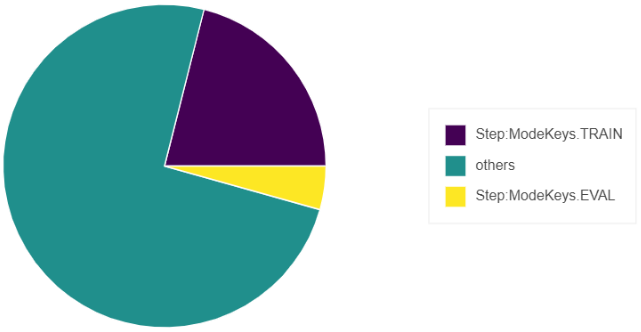
#	node	metric	unit	max	p99	p95	p50	min
0	algo-1	Network	bytes	102429600.84	63.99	0	0	0
1	algo-1	GPU	percentage	94	92.44	89	35	0
2	algo-1	CPU	percentage	100	98.47	95.38	40.3	9.45
3	algo-1	CPU memory	percentage	29.03	27.24	24.56	23.9	5.24
4	algo-1	GPU memory	percentage	85	84	81.2	35	0
5	algo-1	I/O	percentage	51.35	24.48	15.36	0	0

Framework metrics summary

The following two pie charts show the time spent on the TRAIN phase, the EVAL phase, and others. The 'others' includes the time spent between steps (after one step the next step has started). Ideally, most of the training time should be spent on the TRAIN and EVAL phases. If TRAIN/EVAL were not specified in the training script, ste GLOBAL. Your training job spent quite a significant amount of time (74.59%) in phase "others". You should check what is happening in between the steps.

The ratio between the time spent on the TRAIN/EVAL phase and others

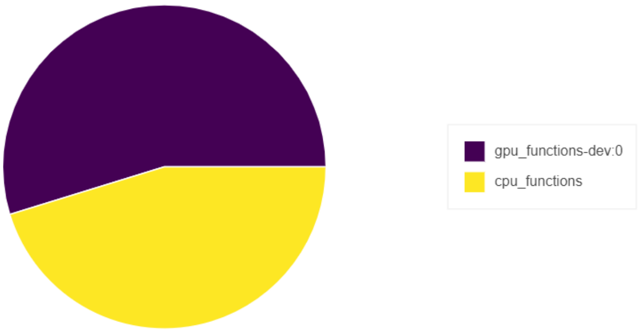
(http



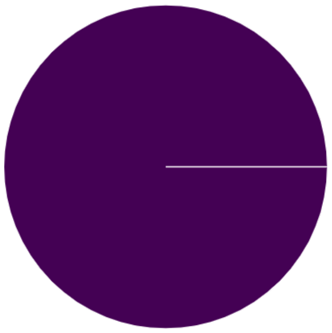
The following piechart shows a breakdown of the CPU/GPU operators. It shows that 54% of training time was spent on executing the "gpu_functions-dev:0" operator.

The ratio between the time spent on CPU/GPU operators

(http



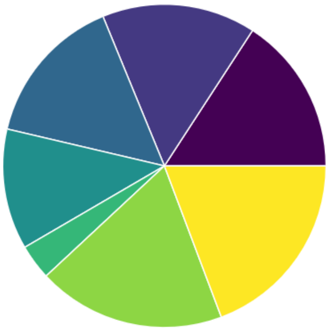
General framework operations



Overview: CPU operators

The following table shows a list of operators that ran on the CPUs. The most expensive operator on the CPUs was "copy_" with 19 %.

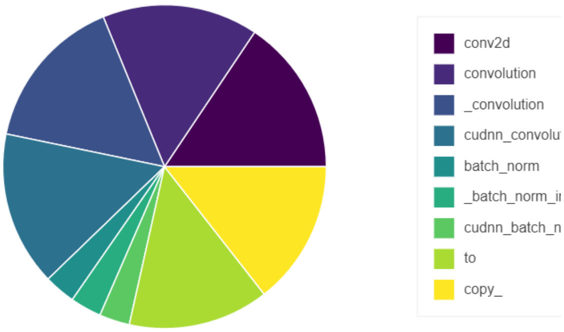
#	Percentage	Cumulative time in microseconds	CPU operator
0	19.27	1989289	copy_
1	18.8	1941327	to
2	15.82	1633363	conv2d
3	15.39	1588949	convolution
4	15.12	1561302	_convolution
5	12.07	1246443	cudnn_convolution
6	3.53	364702	batch_norm



Overview: GPU operators

The following table shows a list of operators that your training job ran on GPU. The most expensive operator on GPU was "conv2d" with 15 %

#	Percentage	Cumulative time in microse	GPU operator
0	15.6	2083446	conv2d
1	15.58	2081440	convolution
2	15.57	2079509	_convolution
3	15.46	2064166	cudnn_convolution
4	14.43	1927845	copy_
5	14.06	1878296	to
6	3.14	418799	batch_norm
7	3.12	417031	_batch_norm_impl_index
8	3.03	405139	cudnn_batch_norm



Rules summary

The following table shows a profiling summary of the Debugger built-in rules. The table is sorted by the rules that triggered the most frequently. During your GPUMemoryIncrease rule was the most frequently triggered. It processed 2657 datapoints and was triggered 252 times.

	Description	Recommendation	Number of times rule triggered	Number of datapoints	
GPUMemoryIncrease	Measures the average GPU memory footprint and triggers if there is a large increase.	Choose a larger instance type with more memory if footprint is close to maximum available memory.	252	2657	
StepOutlier	Detects outliers in step duration. The step duration for forward and backward pass should be roughly the same throughout the training. If there are significant outliers, it may indicate a system stall or bottleneck issues.	Check if there are any bottlenecks (CPU, I/O) correlated to the step outliers.	17	2707	
LowGPUUtilization	Checks if the GPU utilization is low or fluctuating. This can happen due to bottlenecks, blocking calls for synchronizations, or a small batch size.	Check if there are bottlenecks, minimize blocking calls, change distributed training strategy, or increase the batch size.	14	2657	
BatchSize	Checks if GPUs are underutilized because the batch size is too small. To detect this problem, the rule analyzes the average GPU memory footprint, the CPU and the GPU utilization.	The batch size is too small, and GPUs are underutilized. Consider running on a smaller instance type or increasing the batch size.	0	2656	gpu_m
LoadBalancing	Detects workload balancing issues across GPUs. Workload imbalance can occur in training jobs with data parallelism. The gradients are accumulated on a primary GPU, and this GPU might be overused with regard to other GPUs, resulting in reducing the efficiency of data parallelization.	Choose a different distributed training strategy or a different distributed training framework.	0	2657	
Dataloader	Checks how many data loaders are running in parallel and whether the total number is equal the number of available CPU cores. The rule triggers if number is much smaller or larger than the number of available cores. If too small, it might lead to low GPU utilization. If too large, it might impact other compute intensive operations on CPU.	Change the number of data loader processes.	0	0	
CPUBottleneck	Checks if the CPU utilization is high and the GPU utilization is low. It might indicate CPU bottlenecks, where the GPUs are waiting for data to arrive from the CPUs. The rule evaluates the CPU and GPU utilization rates, and triggers the issue if the time spent on the CPU bottlenecks exceeds a threshold percent of the total training time. The default threshold is 50 percent.	Consider increasing the number of data loaders or applying data pre-fetching.	0	2677	
IOBottleneck	Checks if the data I/O wait time is high and the GPU utilization is low. It might indicate IO bottlenecks where GPU is waiting for data to arrive from storage. The rule evaluates the I/O and GPU utilization rates and triggers the issue if the time spent on the IO bottlenecks exceeds a threshold percent of the total training time. The default threshold is 50 percent.	Pre-fetch data or choose different file formats, such as binary formats that improve I/O performance.	0	2677	
MaxInitializationTime	Checks if the time spent on initialization exceeds a threshold percent of the total training time. The rule waits until the first step of training loop starts. The initialization can take longer if downloading the entire dataset from Amazon S3 in File mode. The default threshold is 20 minutes.	Initialization takes too long. If using File mode, consider switching to Pipe mode in case you are using TensorFlow framework.	0	2707	

Analyzing the training loop

Step duration analysis

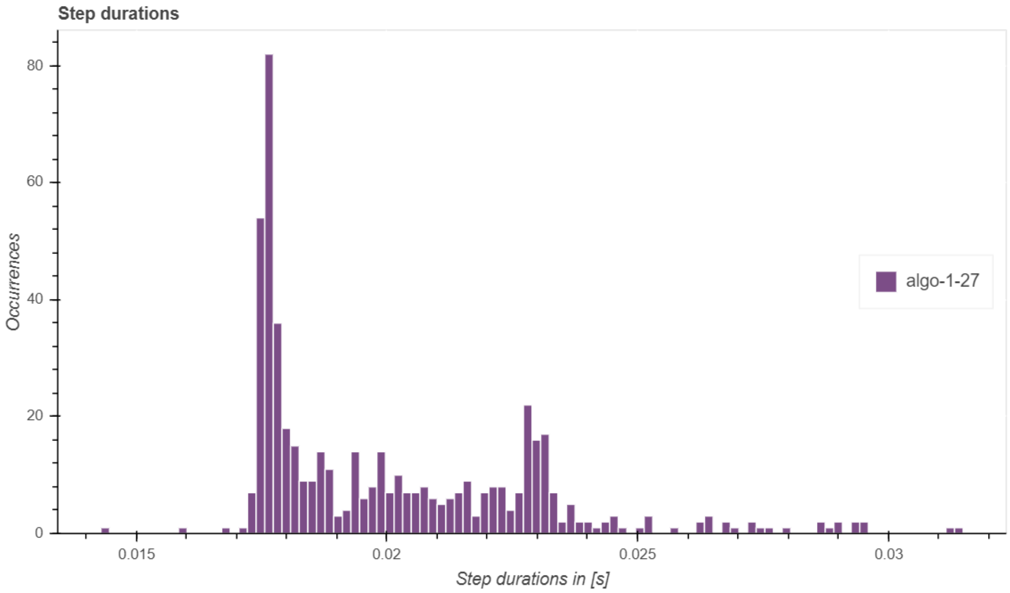
The StepOutlier rule measures step durations and checks for outliers. The rule returns True if duration is larger than 3 times the standard deviation. The rule also takes the parameter mode, that specifies whether steps from training or validation phase should be checked. In your processing job mode was specified as None. Typically the first step is taking significantly more time and to avoid the rule triggering immediately, one can use n_outliers to specify the number of outliers to ignore. n_outliers was set to 10. The rule analysed 2707 datapoints and triggered 17 times.

Step durations on node algo-1-27:

The following table is a summary of the statistics of step durations measured on node algo-1-27. The rule has analyzed the step duration from Step:ModeKey average step duration on node algo-1-27 was 0.05s. The rule detected 2 outliers, where step duration was larger than 3 times the standard deviation of 0.56s

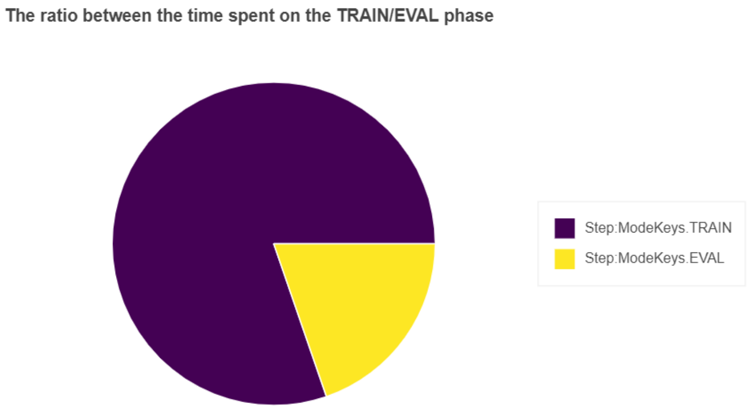
	mean	max	p99	p95	p50	min
Step Durations in [s]	0.05	9.17	0.03	0.03	0.02	0.01

The following histogram shows the step durations measured on the different nodes. You can turn on or turn off the visualization of histograms by selecting c labels in the legend.

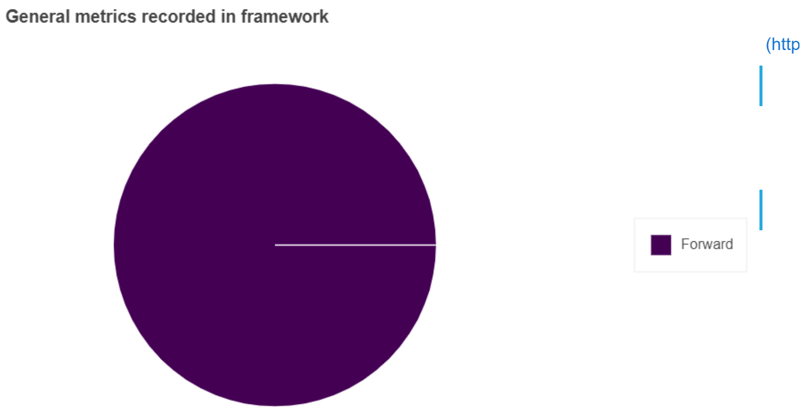


(http

To get a better understanding of what may have caused those outliers, we correlate the timestamps of step outliers with other framework metrics that happened at the same time. The left chart shows how much time was spent in the different framework metrics aggregated by event phase. The chart on the right shows the histogram of normal step durations (without outliers). The following chart shows how much time was spent in the different framework metrics when step outliers occurred. In this chart framework metrics are not aggregated byphase. The chart (in the middle) shows whether step outliers mainly happened during TRAIN or EVAL phase.



(http



GPU utilization analysis

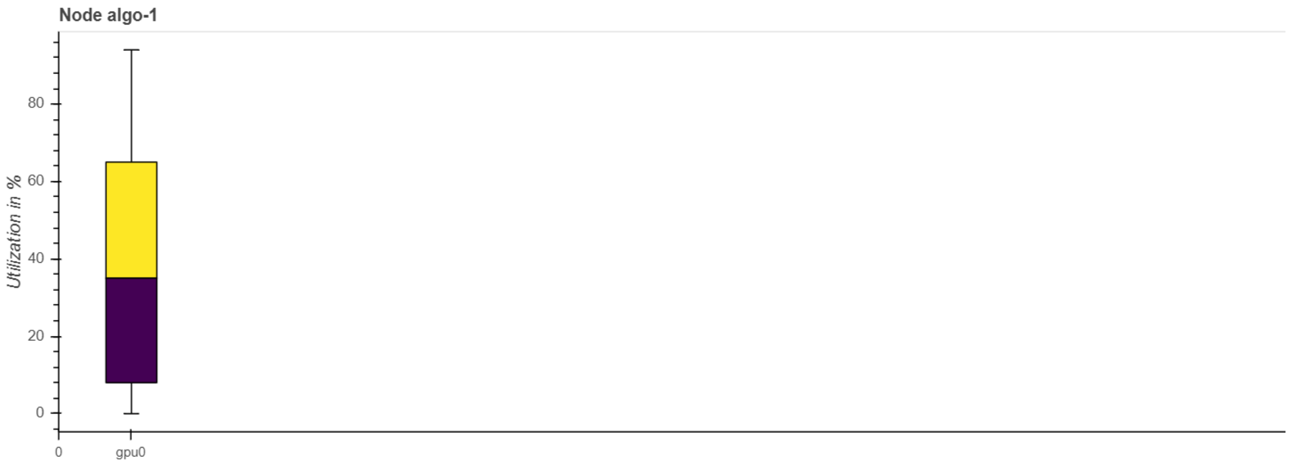
Usage per GPU

The LowGPUUtilization rule checks for a low and fluctuating GPU usage. If the GPU usage is consistently low, it might be caused by bottlenecks or a small batch size. If usage is heavily fluctuating, it can be due to bottlenecks or blocking calls. The rule computed the 95th and 5th percentile of GPU utilization on 500 continuous datapoints and found 14 cases where p95 was above 70% and p5 was below 10%. If p95 is high and p5 is low, it might indicate that the GPU usage is highly fluctuating. If both values are very low, it would mean that the machine is underutilized. During initialization, the GPU usage is likely zero, so the rule skipped the first 1000 data points. The rule analysed 2657 datapoints and triggered 14 times.

Your training job is underutilizing the instance. You may want to consider to either switch to a smaller instance type or to increase the batch size. The last time that the LowGPUUtilization rule was triggered in your training job was on 02/10/2023 at 05:24:00. The following boxplots are a snapshot from the timestamps. They show the utilization per GPU (without outliers). To get a better understanding of the workloads throughout the whole training, you can check the workload histogram in the next section.

GPU utilization of gpu0 on node algo-1:

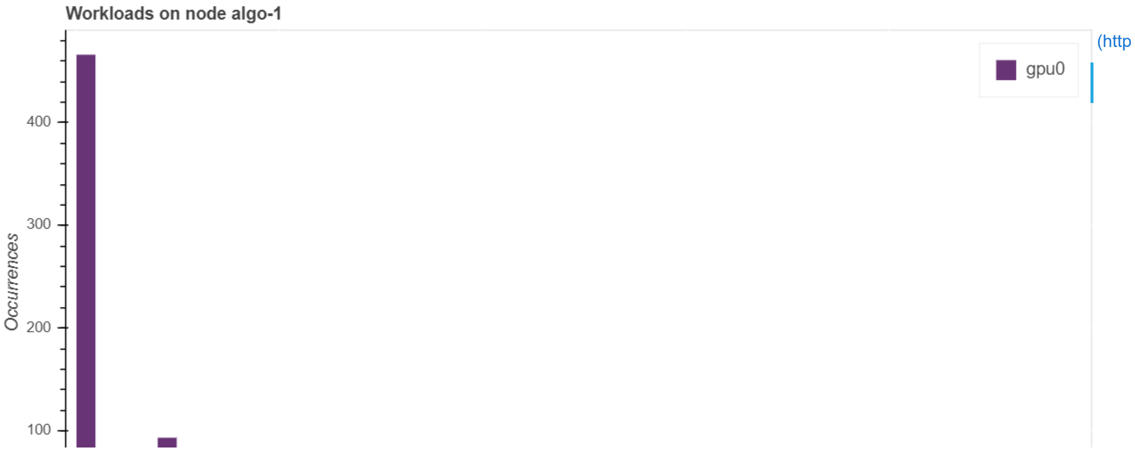
The max utilization of gpu0 on node algo-1 was 94.0% and the 5th percentile was only 3.0% The difference between 5th percentile 3.0% and 95th percentile 88.04999999999995% is quite significant, which means that utilization on gpu0 is fluctuating quite a lot.



Workload balancing

The LoadBalancing rule helps to detect issues in workload balancing between multiple GPUs. It computes a histogram of GPU utilization values for each GPU and compares then the similarity between histograms. The rule checked if the distance of histograms is larger than the threshold of 0.2. During initialization utilization is likely zero, so the rule skipped the first 1000 data points.

The following histogram shows the workload per GPU on node algo-1. You can enable/disable the visualization of a workload by clicking on the label in the legend. Your training job only used one GPU so there is no workload balancing issue.



Model Deploying

In [25…

```
from sagemaker.model_monitor import DataCaptureConfig

data_capture_config = DataCaptureConfig(
    ## TODO: Set config options
    enable_capture = True,
    sampling_percentage=100,
    destination_s3_uri=f"s3://{bucket}/data_capture"
)
```

In [26…

```
# TODO: Deploy your model to an endpoint
predictor = estimator.deploy(initial_instance_count=1, instance_type="ml.m5.xlarge", data_capture_config=data_capture_confi
g)
```

```
------!
```

In [29…

```
from sagemaker.pytorch import PyTorchModel
from sagemaker.predictor import Predictor

#Below is the s3 location of our saved model that was trained by the training job using the best hyperparameters
model_data_artifacts = "s3://sagemaker-us-east-1-910293992207/pytorch-training-2023-02-10-05-01-23-323/output/model.tar.gz"

#We need to define the serializer and deserializer that we will be using as default for our Prediction purposes
jpeg_serializer = sagemaker.serializers.IdentitySerializer("image/jpeg")
json_deserializer = sagemaker.deserializers.JSONDeserializer()

#If we need to override the serializer and deserializer then we need to pass them in an class inheriting the Predictor class
and pass this class as parameter to our PyTorchModel
class ImgPredictor(Predictor):
    def __init__( self, endpoint_name, sagemaker_session):
        super( ImgPredictor, self).__init__(
            endpoint_name,
            sagemaker_session = sagemaker_session,
            serializer = jpeg_serializer,
            deserializer = json_deserializer
        )

pytorch_model = PyTorchModel( model_data = model_data_artifacts,
                            role = role,
                             entry_point= "endpoint_inference.py",
                             py_version = "py36",
                             framework_version = "1.6",
                            predictor_cls = ImgPredictor

                            )

predictor = pytorch_model.deploy( initial_instance_count = 1, instance_type = "ml.t2.medium", data_capture_config=data_captu
re_config)
```

```
---------!
```

Invocation of Endpoint of alphabet: A

In [57…

```
path = "./asl_alphabet/test/A"
for a in range(1,10):
    m = "A"+str(a)+".jpg"
    d = os.path.join(path, m)
    l = "'"+d+"'"
    with open(d , "rb") as f:
        payload = f.read()
        print(f"Expected category no : A")
        response = predictor.predict(payload, initial_args={"ContentType": "image/jpeg"})
        #print(f"Response: {response}")
        predicted_category = np.argmax(response,1) + 1
        print(predicted_category)
```

```
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[1]
Expected category no : A
[2]
Expected category no : A
[20]
```

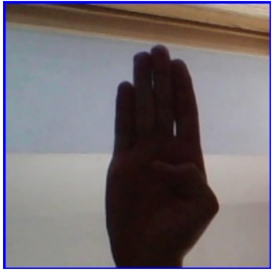
Visualization of the test data samples

In [32…

```
#Testing the deployed endpoint using some test images
#Solution 1: Using the Predictor object directly.
from PIL import Image
import io
import os
import numpy as np

test_dir = "./asl_alphabet/test/"
#test_images = ["D/D10.jpg", "C/C30.jpg", "B/B11.jpg", "A/A10.jpg"]
test_images = ["A/A5.jpg", "B/B11.jpg","C/C30.jpg", "D/D10.jpg"]
test_images_expected_output = ['A', 'B', 'C', 'D' ]
for index in range(len(test_images) ):
    test_img = test_images[index]
    expected_category = test_images_expected_output[index]
    print(f"Test image no: {index+1}")
    test_file_path = os.path.join(test_dir,test_img)
    with open(test_file_path , "rb") as f:
        payload = f.read()
        print("Below is the image that we will be testing:")
        display(Image.open(io.BytesIO(payload)))
        print(f"Expected category no : {expected_category}")
        response = predictor.predict(payload, initial_args={"ContentType": "image/jpeg"})
        print(f"Response: {response}")
        predicted_category = np.argmax(response,1) + 1 #We need to do plus 1 as index starts from zero and prediction is zer
o-indexed
        if predicted_category == 4:
            print(f"Response/Inference for the above image is : 'D'")
        print("------------------------------------------------------------------------")
        if predicted_category == 3:
            print(f"Response/Inference for the above image is : 'C'")
        print("------------------------------------------------------------------------")
        if predicted_category == 2:
            print(f"Response/Inference for the above image is : 'B'")
        print("------------------------------------------------------------------------")
        if predicted_category == 1:
            print(f"Response/Inference for the above image is : 'A'")
        print("------------------------------------------------------------------------")
```

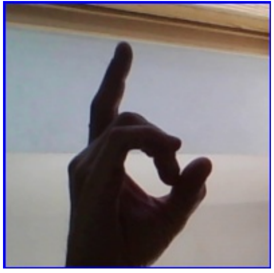
```
Test image no: 1
Below is the image that we will be testing:
```

```
Expected category no : A
Response: [[7.688182353973389, 3.9646029472351074, 0.9547290802001953, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.5011566877365112, 0.0, 3.56672835
3500366, 2.2505133152008057, 0.0, 0.0, 0.0, 0.0, 0.310312956571579, 4.681624412536621, 4.327795505523682, 0.0, 0.0, 0.0, 4.6362524032592
77, 2.499082326889038, 0.0, 0.0, 0.0, 2.8500635623931885, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
Response/Inference for the above image is : 'A'
----------------------------------------------------------------------
Test image no: 2
Below is the image that we will be testing:
```



```
Expected category no : B
Response: [[0.0, 10.367291450500488, 0.0, 0.0, 0.0, 0.0, 0.0, 0.598381757736206, 0.0, 0.8182090520858765, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 1.3108114004135132, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]
----------------------------------------------------------------------
----------------------------------------------------------------------
Response/Inference for the above image is : 'B'
----------------------------------------------------------------------
----------------------------------------------------------------------
Test image no: 3
Below is the image that we will be testing:
```

```
Expected category no : C
Response: [[0.0, 0.0, 10.699620246887207, 3.797069549560547, 0.0, 0.0, 0.0, 0.4872783422470093, 0.0, 0.0, 0.0, 5.143039703369141, 0.0,
0.0, 0.0, 0.3721250891685486, 1.5564589500427246, 0.0, 0.0, 1.8282265663146973, 0.0, 0.0, 0.0, 0.7803218960762024, 0.0, 0.0, 0.0, 0.0,
4.661263942718506, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]
----------------------------------------------------------------------
Response/Inference for the above image is : 'C'
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
Test image no: 4
Below is the image that we will be testing:
```



```
Expected category no : D
Response: [[0.0, 0.0, 3.358764410018921, 8.803892135620117, 0.0, 2.419361114501953, 0.0, 0.39468228816986084, 0.0, 3.692802667617798, 0.
3554805815219879, 0.0, 0.0, 0.0, 0.21633651852607727, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.438459694385528
56, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]]
Response/Inference for the above image is : 'D'
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
```

We have taken 4 test samples, and our model has righly predicted the results.

Data capture

In [58…

```
from sagemaker.s3 import S3Downloader

# In S3 your data will be saved to a datetime-aware path
# Find a path related to a datetime you're interested in
data_path = "s3://sagemaker-us-east-1-910293992207/data_capture/pytorch-inference-2023-02-10-05-41-51-640/AllTraffic/2023/0
2/10/05/"

S3Downloader.download(data_path, "captured_data")

# Feel free to repeat this multiple times and pull in more data
```

In [59…

```
data_path = "s3://sagemaker-us-east-1-910293992207/data_capture/pytorch-inference-2023-02-10-05-41-51-640/AllTraffic/2023/0
2/10/06/"

S3Downloader.download(data_path, "captured_data")
```

In [60…

```
data_path = "s3://sagemaker-us-east-1-910293992207/data_capture/pytorch-inference-2023-02-10-05-41-51-640/AllTraffic/2023/0
2/10/07"

S3Downloader.download(data_path, "captured_data")
```

In [41…

```
!pip install jsonlines
import jsonlines
```

```
Keyring is skipped due to an exception: 'keyring.backends'
Collecting jsonlines
  Downloading jsonlines-3.1.0-py3-none-any.whl (8.6 kB)
Requirement already satisfied: attrs>=19.2.0 in /opt/conda/lib/python3.7/site-packages (from jsonlines) (22.1.0)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.7/site-packages (from jsonlines) (4.4.0)
Installing collected packages: jsonlines
Successfully installed jsonlines-3.1.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It i
s recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip available: 22.3.1 -> 23.0
[notice] To update, run: pip install --upgrade pip
```

In [42…

```
import os

# List the file names we downloaded
file_handles = os.listdir("./captured_data")

# Dump all the data into an array
json_data = []
for jsonl in file_handles:
    with jsonlines.open(f"./captured_data/{jsonl}") as f:
        json_data.append(f.read())
```

In [ ]:

In [38…

```
print(predictor.endpoint_name)
endpoint_name = predictor.endpoint_name
```

```
pytorch-inference-2023-02-10-05-41-51-640
```

In [62…

```
# TODO: Remember to shutdown/delete your endpoint once your work is done
predictor.delete_endpoint()
```

In [ ]: