In [19…
```
# 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)
Requirement already satisfied: protobuf>=3.6.0 in /opt/conda/lib/python3.7/site-packages (from smdebug) (3.20.3)
Requirement already satisfied: boto3>=1.10.32 in /opt/conda/lib/python3.7/site-packages (from smdebug) (1.26.24)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-packages (from smdebug) (20.1)
Collecting pyinstrument==3.4.2
  Using cached pyinstrument-3.4.2-py2.py3-none-any.whl (83 kB)
Requirement already satisfied: numpy>=1.16.0 in /opt/conda/lib/python3.7/site-packages (from smdebug) (1.21.6)
Collecting pyinstrument-cext>=0.2.2
  Using cached pyinstrument_cext-0.2.4-cp37-cp37m-manylinux2010_x86_64.whl (20 kB)
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: 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: 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: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (from packaging->smdebug) (2.4.6)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from packaging->smdebug) (1.14.0)
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)
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)
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 [2]:
```
# 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-1-804209765268
```

Dataset

The dataset we used for this project is dogs_vs_cats dataset that can be found in the link (https://www.kaggle.com/datasets/salader/dogs-vs-cats) It comprises: -DogCatImages

- Train folder
  -- Dogs (10000 images)
  -- Cats (10000 images)
- Test folder
  -- Dogs (2500 images)
  -- Cats (2500 images)

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 974.5 kB/s eta 0:00:00a 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 1.2 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=02ca78bede098c74e20a94cfa944981928f42879d3bf20154f
0baf8b2ac983db
  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 = "username"
kaggle_key = "kaggle 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 [7]:
```
!kaggle datasets download -d salader/dogs-vs-cats
```

```
Downloading dogs-vs-cats.zip to /root/deep_learning
 98%|████████████████████████████████▉  | 1.05G/1.06G [00:05<00:00, 232MB/s]
100%|█████████████████████████████████| 1.06G/1.06G [00:12<00:00, 93.0MB/s]
```

In [8]:
```
# unzipping dogs_vs_cats.zip file using quiet
!unzip -q dogs-vs-cats.zip # -q for quiet
```

In [ ]:
```
```

In [9]:
```
# uploading data in AWS S3
prefix ="DogCatImages"
print("Starting to upload DogCatImages")

inputs = sagemaker_session.upload_data(path="dogs_vs_cats", bucket=bucket, key_prefix=prefix)
print(f"Input path ( S3 file path ): {inputs}")
```

```
Starting to upload dogImages
Input path ( S3 file path ): s3://sagemaker-us-east-1-804209765268/DogCatImages
```

In [4]:
```
inputs = "s3://sagemaker-us-east-1-804209765268/DogCatImages"
print(f"Input path ( S3 file path ): {inputs}")
```

```
Input path ( S3 file path ): s3://sagemaker-us-east-1-804209765268/DogCatImages
```

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 [6]:
```
#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 [7]:
```
from  sagemaker.pytorch import PyTorch

estimator = PyTorch(
    entry_point = "hpo.py",
    base_job_name = "dog-cat-classification-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 [8]:
```
# 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 [9]:
```
# Get the best estimators and the best HPs

best_estimator = tuner.best_estimator()

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

```
2023-02-03 05:37:04 Starting - Found matching resource for reuse
2023-02-03 05:37:04 Downloading - Downloading input data
2023-02-03 05:37:04 Training - Training image download completed. Training in progress.
2023-02-03 05:37:04 Uploading - Uploading generated training model
2023-02-03 05:37:04 Completed - Resource retained for reuse
```

Out[9]:
```
{'_tuning_objective_metric': '"average test loss"',
 'batch_size': '"128"',
 'eps': '7.055629880899328e-09',
 'lr': '0.0013884806300905848',
 'sagemaker_container_log_level': '20',
 'sagemaker_estimator_class_name': '"PyTorch"',
 'sagemaker_estimator_module': '"sagemaker.pytorch.estimator"',
 'sagemaker_job_name': '"dog-cat-classification-hpo-2023-02-03-04-56-49-111"',
 'sagemaker_program': '"hpo.py"',
 'sagemaker_region': '"us-east-1"',
 'sagemaker_submit_directory': '"s3://sagemaker-us-east-1-804209765268/dog-cat-classification-hpo-2023-02-03-04-56-49-111/source/sourced
ir.tar.gz"',
 'weight_decay': '0.007288495752729303'}
```

In [15…
```
# Below are the hyperparameters markdown, that can be used instead of re-running the entire hypertuning job
```

In [11…
```
##################
{'batch_size': 128, 'eps': '7.055629880899328e-09', 'lr': '0.0013884806300905848', 'weight_decay': '0.007288495752729303'}
```

Out[11]:
```
{'batch_size': 128,
 'eps': '7.055629880899328e-09',
 'lr': '0.0013884806300905848',
 'weight_decay': '0.007288495752729303'}
```

In [12…
```
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': 128, 'eps': '7.055629880899328e-09', 'lr': '0.0013884806300905848', 'weight_decay': '0.007288495752729303'}
```

Model Profiling and Debugging

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

In [14…
```
# Setting up debugger and profiler rules and configs
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 [15…
```
# 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#de
bugger-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)
```

```
2023-02-03 05:44:05 Starting - Starting the training job...
2023-02-03 05:44:34 Starting - Preparing the instances for trainingVanishingGradient: InProgress
Overfit: InProgress
Overtraining: InProgress
PoorWeightInitialization: InProgress
ProfilerReport: InProgress
.........
2023-02-03 05:45:55 Downloading - Downloading input data...
2023-02-03 05:46:36 Training - Downloading the training image......
2023-02-03 05:47:36 Training - Training image download completed. Training in progress....bash: cannot set terminal process group (-1):
Inappropriate ioctl for device
bash: no job control in this shell
2023-02-03 05:47:51,458 sagemaker-training-toolkit INFO     Imported framework sagemaker_pytorch_container.training
2023-02-03 05:47:51,486 sagemaker_pytorch_container.training INFO     Block until all host DNS lookups succeed.
2023-02-03 05:47:51,488 sagemaker_pytorch_container.training INFO     Invoking user training script.
2023-02-03 05:47:51,721 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": 128,
        "eps": "7.055629880899328e-09",
        "lr": "0.0013884806300905848",
        "weight_decay": "0.007288495752729303"
    },
    "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-2023-02-03-05-44-04-605",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/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":128,"eps":"7.055629880899328e-09","lr":"0.0013884806300905848","weight_decay":"0.007288495752729303"}
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-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/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":128,"eps":"7.055629
880899328e-09","lr":"0.0013884806300905848","weight_decay":"0.007288495752729303"},"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-2023-02-03-05-44-04-605","log_level":20,"master_hostname":"algo-1","model_dir":"/op
t/ml/model","module_dir":"s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/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","128","--eps","7.055629880899328e-09","--lr","0.0013884806300905848","--weight_decay","0.00728849575272930
3"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_BATCH_SIZE=128
SM_HP_EPS=7.055629880899328e-09
SM_HP_LR=0.0013884806300905848
SM_HP_WEIGHT_DECAY=0.007288495752729303
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 128 --eps 7.055629880899328e-09 --lr 0.0013884806300905848 --weight_decay 0.0072884
95752729303
[2023-02-03 05:47:52.433 algo-1:27 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2023-02-03 05:47:52.630 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: 0.0013884806300905848,  Eps: 7.055629880899328e-09, Weight-decay: 0.007288495752729303, Batch Size: 128, Epoch: 2
Data Dir Path: /opt/ml/input/data/train
Model Dir  Path: /opt/ml/model
Output Dir  Path: /opt/ml/output/data
[2023-02-03 05:47:56.538 algo-1:27 INFO json_config.py:91] Creating hook from json_config at /opt/ml/input/config/debughookconfig.json.
[2023-02-03 05:47:56.540 algo-1:27 INFO hook.py:199] tensorboard_dir has not been set for the hook. SMDebug will not be exporting tensor
board summaries.
[2023-02-03 05:47:56.541 algo-1:27 INFO hook.py:253] Saving to /opt/ml/output/tensors
[2023-02-03 05:47:56.541 algo-1:27 INFO state_store.py:77] The checkpoint config file /opt/ml/input/config/checkpointconfig.json does no
t exist.
[2023-02-03 05:47:56.568 algo-1:27 INFO hook.py:584] name:fc.0.weight count_params:524288
[2023-02-03 05:47:56.569 algo-1:27 INFO hook.py:584] name:fc.0.bias count_params:256
[2023-02-03 05:47:56.569 algo-1:27 INFO hook.py:584] name:fc.2.weight count_params:34048
[2023-02-03 05:47:56.569 algo-1:27 INFO hook.py:584] name:fc.2.bias count_params:133
[2023-02-03 05:47:56.570 algo-1:27 INFO hook.py:586] Total Trainable Params: 558725
Epoch 1 - Starting Training phase.
Epoch: 1 - Training Model on Complete Training Dataset!
[2023-02-03 05:47:58.396 algo-1:27 INFO hook.py:413] Monitoring the collections: relu_input, losses, gradients, CrossEntropyLoss_output_
0
[2023-02-03 05:47:58.397 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-1675403272631281.5_train-0-stepstart-1675403278397168.8/python_stats.
[2023-02-03 05:47:58.409 algo-1:27 INFO hook.py:476] Hook is writing from the hook with pid: 27
[2023-02-03 05:48:20.241 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-1675403278407677.2_train-0-forwardpassend-1675403300214184.8/python_stats.
[2023-02-03 05:48:21.530 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-1675403300246165.2_train-1-stepstart-1675403301526305.0/python_stats.
[2023-02-03 05:48:26.710 algo-1:27 INFO python_profiler.py:182] Dumping cProfile stats to /opt/ml/output/profiler/framework/pytorch/cpro
file/27-algo-1/train-1-stepstart-1675403301533753.0_train-1-forwardpassend-1675403306709718.8/python_stats.
[2023-02-03 05:48:27.837 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-1675403306714893.0_train-2-stepstart-1675403307837118.2/python_stats.
[2023-02-03 05:48:32.472 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-1675403307840172.2_train-2-forwardpassend-1675403312471826.8/python_stats.
[2023-02-03 05:48:33.602 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-1675403312474057.2_train-3-stepstart-1675403313602232.8/python_stats.
[2023-02-03 05:48:38.273 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-1675403313605978.8_train-3-forwardpassend-1675403318273472.5/python_stats.
[2023-02-03 05:48:39.358 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-1675403318275233.0_train-4-stepstart-1675403319358015.5/python_stats.
[2023-02-03 05:48:44.018 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-1675403319361138.0_train-4-forwardpassend-1675403324017898.2/python_stats.
[2023-02-03 05:48:45.168 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-1675403324019744.2_train-5-stepstart-1675403325167789.5/python_stats.
[2023-02-03 05:48:49.816 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-1675403325171156.0_train-5-forwardpassend-1675403329816549.0/python_stats.
[2023-02-03 05:48:50.994 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-1675403329818229.5_train-6-stepstart-1675403330993829.0/python_stats.
[2023-02-03 05:48:55.706 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-1675403331028455.0_train-6-forwardpassend-1675403335706467.8/python_stats.
[2023-02-03 05:48:56.814 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-1675403335708371.8_train-7-stepstart-1675403336814075.0/python_stats.
[2023-02-03 05:49:00.958 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-1675403336817048.5_train-7-forwardpassend-1675403340957813.5/python_stats.
[2023-02-03 05:49:02.025 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-1675403340959658.5_train-8-stepstart-1675403342025404.2/python_stats.
[2023-02-03 05:49:06.148 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-1675403342033070.5_train-8-forwardpassend-1675403346147665.2/python_stats.
[2023-02-03 05:49:07.063 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-1675403346149374.0_train-9-stepstart-1675403347062411.2/python_stats.
[2023-02-03 05:49:11.065 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-1675403347065794.5_train-9-forwardpassend-1675403351064794.5/python_stats.
[2023-02-03 05:49:11.959 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-1675403351066660.8_train-10-stepstart-1675403351958953.0/python_stats.
Train set:  [16000/20000 (80%)]#011 Loss: 0.17#011Accuracy: 14736/16000 (92.10%)
Train set:  [20000/20000 (100%)]#011 Loss: 0.16#011Accuracy: 18525/20000 (92.62%)
Train set: Average loss: 0.1932, Accuracy: 18525/20000 (93%)
Epoch 1 - Starting Testing phase.
Epoch: 1 - Testing Model on Complete Testing Dataset!
Test set: Average loss: 0.1441, Accuracy: 4697/5000 (94%)
```

```
Epoch 2 - Starting Training phase.
Epoch: 2 - Training Model on Complete Training Dataset!
VanishingGradient: InProgress
Overfit: Error
Overtraining: Error
PoorWeightInitialization: InProgress
VanishingGradient: InProgress
Overfit: Error
Overtraining: Error
PoorWeightInitialization: Error
VanishingGradient: Error
Overfit: Error
Overtraining: Error
PoorWeightInitialization: Error
Train set:  [16000/20000 (80%)]#011 Loss: 0.15#011Accuracy: 14999/16000 (93.74%)
Train set:  [20000/20000 (100%)]#011 Loss: 0.05#011Accuracy: 18796/20000 (93.98%)
Train set: Average loss: 0.1402, Accuracy: 18796/20000 (94%)
Epoch 2 - Starting Testing phase.
Epoch: 2 - Testing Model on Complete Testing Dataset!
Test set: Average loss: 0.1635, Accuracy: 4657/5000 (93%)
Starting to Save the Model
Completed Saving the Model
INFO:__main__:Running on Device cuda:0
INFO:__main__:Hyperparameters : LR: 0.0013884806300905848,  Eps: 7.055629880899328e-09, Weight-decay: 0.007288495752729303, Batch Size:
128, 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  8%|▊         | 8.29M/97.8M [00:00<00:01, 86.9MB/s]#015 17%|█▋        | 16.6M/97.8
M [00:00<00:00, 86.9MB/s]#015 26%|██▌       | 25.0M/97.8M [00:00<00:00, 87.1MB/s]#015 34%|███▍      | 33.4M/97.8M [00:00<00:00, 87.4MB/
s]#015 43%|████▎     | 41.7M/97.8M [00:00<00:00, 87.4MB/s]#015 51%|█████     | 50.2M/97.8M [00:00<00:00, 87.8MB/s]#015 60%|█████▉    |
58.6M/97.8M [00:00<00:00, 87.9MB/s]#015 69%|██████▊   | 67.1M/97.8M [00:00<00:00, 88.2MB/s]#015 77%|███████▋  | 75.5M/97.8M [00:00<00:
00, 88.4MB/s]#015 86%|████████▌ | 84.0M/97.8M [00:01<00:00, 88.6MB/s]#015 95%|█████████▍| 92.6M/97.8M [00:01<00:00, 88.8MB/s]#015100%|█
████████| 97.8M/97.8M [00:01<00:00, 88.2MB/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:  [16000/20000 (80%)]#011 Loss: 0.17#011Accuracy: 14736/16000 (92.10%)
INFO:__main__:
Train set:  [20000/20000 (100%)]#011 Loss: 0.16#011Accuracy: 18525/20000 (92.62%)
INFO:__main__:
Train set: Average loss: 0.1932, Accuracy: 18525/20000 (93%)
INFO:__main__:Epoch 1 - Starting Testing phase.
INFO:__main__:Epoch: 1 - Testing Model on Complete Testing Dataset!
INFO:__main__:
Test set: Average loss: 0.1441, Accuracy: 4697/5000 (94%)
INFO:__main__:Epoch 2 - Starting Training phase.
INFO:__main__:Epoch: 2 - Training Model on Complete Training Dataset!
INFO:__main__:
Train set:  [16000/20000 (80%)]#011 Loss: 0.15#011Accuracy: 14999/16000 (93.74%)
INFO:__main__:
Train set:  [20000/20000 (100%)]#011 Loss: 0.05#011Accuracy: 18796/20000 (93.98%)
INFO:__main__:
Train set: Average loss: 0.1402, Accuracy: 18796/20000 (94%)
INFO:__main__:Epoch 2 - Starting Testing phase.
2023-02-03 05:56:19,768 sagemaker-training-toolkit INFO     Reporting training SUCCESS
INFO:__main__:Epoch: 2 - Testing Model on Complete Testing Dataset!
INFO:__main__:
Test set: Average loss: 0.1635, Accuracy: 4657/5000 (93%)
INFO:__main__:Starting to Save the Model
INFO:__main__:Completed Saving the Model

2023-02-03 05:56:33 Uploading - Uploading generated training model
2023-02-03 05:56:54 Completed - Training job completed
Training seconds: 665
Billable seconds: 665
```

In [37…

```
#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 [38…

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

```
Jobname: pytorch-training-2023-02-03-05-44-04-605
Client: <botocore.client.SageMaker object at 0x7f7dd72b75d0>
Description: {'TrainingJobName': 'pytorch-training-2023-02-03-05-44-04-605', 'TrainingJobArn': 'arn:aws:sagemaker:us-east-1:80420976526
8:training-job/pytorch-training-2023-02-03-05-44-04-605', 'ModelArtifacts': {'S3ModelArtifacts': 's3://sagemaker-us-east-1-804209765268/
pytorch-training-2023-02-03-05-44-04-605/output/model.tar.gz'}, 'TrainingJobStatus': 'Completed', 'SecondaryStatus': 'Completed', 'Hyper
Parameters': {'batch_size': '128', 'eps': '"7.055629880899328e-09"', 'lr': '"0.0013884806300905848"', 'sagemaker_container_log_level':
'20', 'sagemaker_job_name': '"pytorch-training-2023-02-03-05-44-04-605"', 'sagemaker_program': '"train_model.py"', 'sagemaker_region':
'"us-east-1"', 'sagemaker_submit_directory': '"s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/source/sou
rcedir.tar.gz"', 'weight_decay': '"0.007288495752729303"'}, 'AlgorithmSpecification': {'TrainingImage': '763104351884.dkr.ecr.us-east-1.
amazonaws.com/pytorch-training:1.6-gpu-py36', 'TrainingInputMode': 'File', 'EnableSageMakerMetricsTimeSeries': True}, 'RoleArn': 'arn:aw
s:iam::804209765268:role/service-role/AmazonSageMaker-ExecutionRole-20230121T124407', 'InputDataConfig': [{'ChannelName': 'train', 'Data
Source': {'S3DataSource': {'S3DataType': 'S3Prefix', 'S3Uri': 's3://sagemaker-us-east-1-804209765268/DogCatImages', 'S3DataDistributionT
ype': 'FullyReplicated'}}, 'CompressionType': 'None', 'RecordWrapperType': 'None'}], 'OutputDataConfig': {'KmsKeyId': '', 'S3OutputPat
h': 's3://sagemaker-us-east-1-804209765268/'}, 'ResourceConfig': {'InstanceType': 'ml.g4dn.xlarge', 'InstanceCount': 1, 'VolumeSizeInG
B': 30}, 'StoppingCondition': {'MaxRuntimeInSeconds': 86400}, 'CreationTime': datetime.datetime(2023, 2, 3, 5, 44, 5, 850000, tzinfo=tzl
ocal()), 'TrainingStartTime': datetime.datetime(2023, 2, 3, 5, 45, 40, 749000, tzinfo=tzlocal()), 'TrainingEndTime': datetime.datetime(2
023, 2, 3, 5, 56, 45, 191000, tzinfo=tzlocal()), 'LastModifiedTime': datetime.datetime(2023, 2, 3, 5, 57, 14, 141000, tzinfo=tzlocal()),
'SecondaryStatusTransitions': [{'Status': 'Starting', 'StartTime': datetime.datetime(2023, 2, 3, 5, 44, 5, 850000, tzinfo=tzlocal()), 'E
ndTime': datetime.datetime(2023, 2, 3, 5, 45, 40, 749000, tzinfo=tzlocal()), 'StatusMessage': 'Preparing the instances for training'},
{'Status': 'Downloading', 'StartTime': datetime.datetime(2023, 2, 3, 5, 45, 40, 749000, tzinfo=tzlocal()), 'EndTime': datetime.datetime
(2023, 2, 3, 5, 46, 31, 288000, tzinfo=tzlocal()), 'StatusMessage': 'Downloading input data'}, {'Status': 'Training', 'StartTime': datet
ime.datetime(2023, 2, 3, 5, 46, 31, 288000, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 3, 5, 56, 24, 641000, tzinfo=tzloca
l()), 'StatusMessage': 'Training image download completed. Training in progress.'}, {'Status': 'Uploading', 'StartTime': datetime.dateti
me(2023, 2, 3, 5, 56, 24, 641000, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 3, 5, 56, 45, 191000, tzinfo=tzlocal()), 'Sta
tusMessage': 'Uploading generated training model'}, {'Status': 'Completed', 'StartTime': datetime.datetime(2023, 2, 3, 5, 56, 45, 19100
0, tzinfo=tzlocal()), 'EndTime': datetime.datetime(2023, 2, 3, 5, 56, 45, 191000, tzinfo=tzlocal()), 'StatusMessage': 'Training job comp
leted'}], 'EnableNetworkIsolation': False, 'EnableInterContainerTrafficEncryption': False, 'EnableManagedSpotTraining': False, 'Training
TimeInSeconds': 665, 'BillableTimeInSeconds': 665, 'DebugHookConfig': {'S3OutputPath': 's3://sagemaker-us-east-1-804209765268/', 'Collec
tionConfigurations': [{'CollectionName': 'gradients', 'CollectionParameters': {'save_interval': '500'}}, {'CollectionName': 'CrossEntrop
yLoss_output_0', 'CollectionParameters': {'eval.save_interval': '1', 'include_regex': 'CrossEntropyLoss_output_0', 'train.save_interva
l': '10'}}, {'CollectionName': 'relu_input', 'CollectionParameters': {'include_regex': '.*relu_input', 'save_interval': '500'}}]}, 'Debu
gRuleConfigurations': [{'RuleConfigurationName': 'VanishingGradient', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.co
m/sagemaker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'VanishingGradient'}}, {'RuleConfiguration
Name': 'Overfit', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.com/sagemaker-debugger-rules:latest', 'VolumeSizeInG
B': 0, 'RuleParameters': {'rule_to_invoke': 'Overfit'}}, {'RuleConfigurationName': 'Overtraining', 'RuleEvaluatorImage': '503895931360.d
kr.ecr.us-east-1.amazonaws.com/sagemaker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'Overtrainin
g'}}, {'RuleConfigurationName': 'PoorWeightInitialization', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-east-1.amazonaws.com/sagemake
r-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'PoorWeightInitialization'}}], 'DebugRuleEvaluationS
tatuses': [{'RuleConfigurationName': 'VanishingGradient', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:804209765268:processing-j
ob/pytorch-training-2023-02-0-vanishinggradient-eafaea77', 'RuleEvaluationStatus': 'Error', 'StatusDetails': 'InternalServerError: We en
countered an internal error. Please try again.', 'LastModifiedTime': datetime.datetime(2023, 2, 3, 5, 54, 40, 99000, tzinfo=tzlocal())},
{'RuleConfigurationName': 'Overfit', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:804209765268:processing-job/pytorch-training-2
023-02-0-overfit-cb5fbbd3', 'RuleEvaluationStatus': 'Error', 'StatusDetails': 'InternalServerError: We encountered an internal error. Pl
ease try again.', 'LastModifiedTime': datetime.datetime(2023, 2, 3, 5, 54, 40, 99000, tzinfo=tzlocal())}, {'RuleConfigurationName': 'Over
training', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:804209765268:processing-job/pytorch-training-2023-02-0-overtraining-3f0
55a04', 'RuleEvaluationStatus': 'Error', 'StatusDetails': 'InternalServerError: We encountered an internal error. Please try again.', 'L
astModifiedTime': datetime.datetime(2023, 2, 3, 5, 54, 40, 99000, tzinfo=tzlocal())}, {'RuleConfigurationName': 'PoorWeightInitializatio
n', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:804209765268:processing-job/pytorch-training-2023-02-0-poorweightinitialization
-2fde8418', 'RuleEvaluationStatus': 'Error', 'StatusDetails': 'InternalServerError: We encountered an internal error. Please try agai
n.', 'LastModifiedTime': datetime.datetime(2023, 2, 3, 5, 54, 40, 99000, tzinfo=tzlocal())}], 'ProfilerConfig': {'S3OutputPath': 's3://s
agemaker-us-east-1-804209765268/', 'ProfilingIntervalInMilliseconds': 500, 'ProfilingParameters': {'DataloaderProfilingConfig': '{"Start
Step": 0, "NumSteps": 10, "MetricsRegex": ".*", }', 'DetailedProfilingConfig': '{"StartStep": 0, "NumSteps": 10, }', 'FileOpenFailThresh
old': '50', 'HorovodProfilingConfig': '{"StartStep": 0, "NumSteps": 10, }', 'LocalPath': '/opt/ml/output/profiler', 'PythonProfilingConf
ig': '{"StartStep": 0, "NumSteps": 10, "ProfilerName": "cprofile", "cProfileTimer": "total_time", }', 'RotateFileCloseIntervalInSecond
s': '60', 'RotateMaxFileSizeInBytes': '10485760', 'SMDataParallelProfilingConfig': '{"StartStep": 0, "NumSteps": 10, }'}, 'DisableProfil
er': False}, 'ProfilerRuleConfigurations': [{'RuleConfigurationName': 'ProfilerReport', 'RuleEvaluatorImage': '503895931360.dkr.ecr.us-e
ast-1.amazonaws.com/sagemaker-debugger-rules:latest', 'VolumeSizeInGB': 0, 'RuleParameters': {'rule_to_invoke': 'ProfilerReport'}}], 'Pr
ofilerRuleEvaluationStatuses': [{'RuleConfigurationName': 'ProfilerReport', 'RuleEvaluationJobArn': 'arn:aws:sagemaker:us-east-1:8042097
65268:processing-job/pytorch-training-2023-02-0-profilerreport-81a94c70', 'RuleEvaluationStatus': 'IssuesFound', 'StatusDetails': 'RuleE
valuationConditionMet: Evaluation of the rule ProfilerReport at step 11 resulted in the condition being met\n', 'LastModifiedTime': date
time.datetime(2023, 2, 3, 5, 57, 14, 136000, tzinfo=tzlocal())}], 'ProfilingStatus': 'Enabled', 'ResponseMetadata': {'RequestId': '40ca4
4c5-e6ad-47aa-b557-c8c56e0ddb1c', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '40ca44c5-e6ad-47aa-b557-c8c56e0ddb1c', 'co
ntent-type': 'application/x-amz-json-1.1', 'content-length': '7188', 'date': 'Fri, 03 Feb 2023 07:56:37 GMT'}, 'RetryAttempts': 0}}
```

In [39…

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

```
[2023-02-03 07:56:47.210 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:18 INFO s3_trial.py:42] Loading trial debug-output at
path s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/debug-output
[2023-02-03 07:56:47.539 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:18 WARNING s3handler.py:183] Encountered the exceptio
n An error occurred while reading from response stream: ('Connection broken: IncompleteRead(0 bytes read, 4808 more expected)', Incomple
teRead(0 bytes read, 4808 more expected)) while reading s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/d
ebug-output/index/000000000/000000000140_worker_0.json . Will retry now
```

In [40…

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

```
[2023-02-03 07:56:54.613 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:18 INFO trial.py:198] Training has ended, will refres
h one final time in 1 sec.
[2023-02-03 07:56:55.636 datascience-1-0-ml-t3-medium-1abf3407f667f989be9d86559395:18 INFO trial.py:210] Loaded all steps
```

Out[40]:
```
['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 [41…
```
len(trial.tensor("CrossEntropyLoss_output_0").steps(mode=ModeKeys.TRAIN))
```

Out[41]:
```
32
```

In [42…
```
len(trial.tensor("CrossEntropyLoss_output_0").steps(mode=ModeKeys.EVAL))
```

Out[42]:
```
80
```

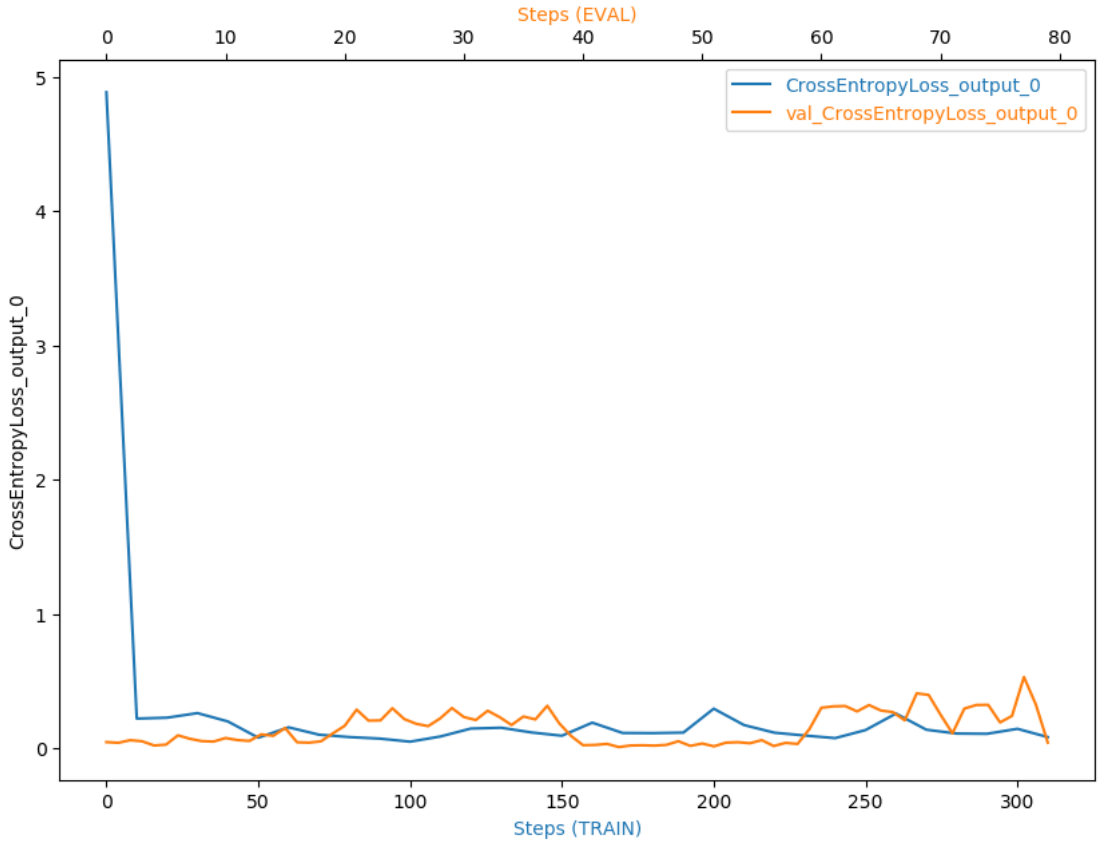
In [43…

```
#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 [44…

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

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



In [30…
```
# 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-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/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 [31…
```
! aws s3 ls {rule_output_path} --recursive
```

```
2023-02-03 05:56:40     406990 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-report.html
2023-02-03 05:56:40     260679 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-report.ipynb
2023-02-03 05:56:35        192 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/Batc
hSize.json
2023-02-03 05:56:35      19092 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/CPUB
ottleneck.json
2023-02-03 05:56:35        126 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/Data
loader.json
2023-02-03 05:56:35        328 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/GPUM
emoryIncrease.json
2023-02-03 05:56:35       3250 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/IOBo
ttleneck.json
2023-02-03 05:56:35        318 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/Load
Balancing.json
2023-02-03 05:56:35        340 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/LowG
PUUtilization.json
2023-02-03 05:56:35        232 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/MaxI
nitializationTime.json
2023-02-03 05:56:35       1354 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/Over
allFrameworkMetrics.json
2023-02-03 05:56:35        622 pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/profiler-reports/Over
allSystemUsage.json
2023-02-03 05:56:35       2377 pytorch-training-2023-02-03-05-44-04-605/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 [32…
```
! aws s3 cp {rule_output_path} ./ --recursive
```

```
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/BatchSize.json to ProfilerReport/profiler-output/profiler-reports/BatchSize.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/GPUMemoryIncrease.json to ProfilerReport/profiler-output/profiler-reports/GPUMemoryIncrease.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/MaxInitializationTime.json to ProfilerReport/profiler-output/profiler-reports/MaxInitializationTime.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-report.ipynb to ProfilerReport/profiler-output/profiler-report.ipynb
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/CPUBottleneck.json to ProfilerReport/profiler-output/profiler-reports/CPUBottleneck.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/Dataloader.json to ProfilerReport/profiler-output/profiler-reports/Dataloader.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/LoadBalancing.json to ProfilerReport/profiler-output/profiler-reports/LoadBalancing.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/IOBottleneck.json to ProfilerReport/profiler-output/profiler-reports/IOBottleneck.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-report.html to ProfilerReport/profiler-output/profiler-report.html
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/LowGPUUtilization.json to ProfilerReport/profiler-output/profiler-reports/LowGPUUtilization.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/OverallFrameworkMetrics.json to ProfilerReport/profiler-output/profiler-reports/OverallFrameworkMetrics.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/OverallSystemUsage.json to ProfilerReport/profiler-output/profiler-reports/OverallSystemUsage.json
download: s3://sagemaker-us-east-1-804209765268/pytorch-training-2023-02-03-05-44-04-605/rule-output/ProfilerReport/profiler-output/prof
iler-reports/StepOutlier.json to ProfilerReport/profiler-output/profiler-reports/StepOutlier.json
```

In [33…
```
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 [34…
```
import IPython

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

Out[34]:

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.

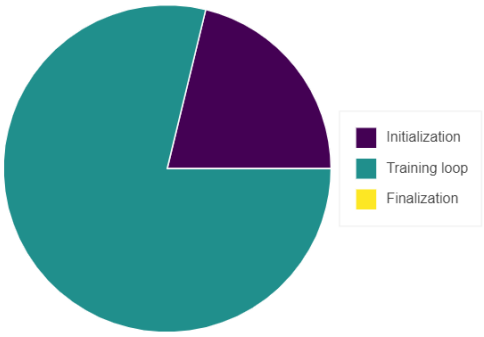
In [4]:

```
# Parameters
processing_job_arn = "arn:aws:sagemaker:us-east-1:804209765268:processing-job/pytorch-training-2023-02-0-profilerreport-81a94
```

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 02/03/2023 at 05:45:44 and ran for 615 seconds.

#		Job Statistics
0	Start time	05:45:44 02/03/2023
1	End time	05:55:59 02/03/2023
2	Job duration	615 seconds
3	Training loop start	05:47:58 02/03/2023
4	Training loop end	05:55:59 02/03/2023
5	Training loop duration	481 seconds
6	Initialization time	133 seconds
7	Finalization time	0 seconds
8	Initialization	21 %
9	Training loop	78 %
10	Finalization	0 %



System usage statistics

The median total GPU utilization on node algo-1 is 0%. The median total CPU utilization is 47%.

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.

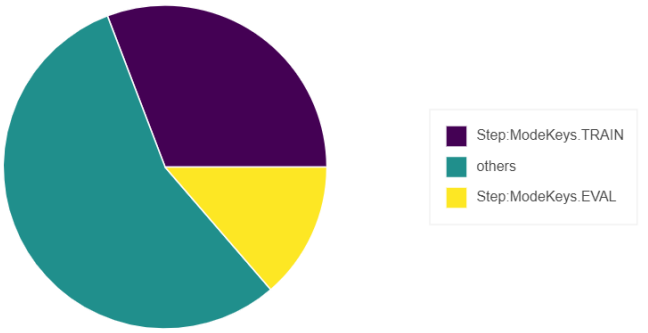
#	node	metric	unit	max	p99	p95	p50	min
0	algo-1	Network	bytes	91849545.82	138.3	0	0	0
1	algo-1	GPU	percentage	100	100	100	0	0
2	algo-1	CPU	percentage	99.51	98.84	90.67	47.61	10.07
3	algo-1	CPU memory	percentage	32.64	29.5	25.58	23.41	4.04
4	algo-1	GPU memory	percentage	100	97	92	0	0
5	algo-1	I/O	percentage	45.24	27.83	22.96	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 (55.53%) 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

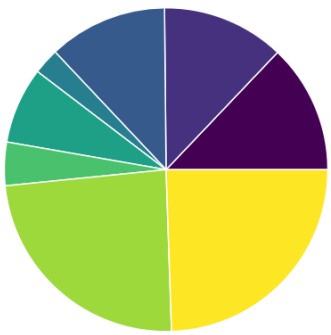
General framework operations

(http

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 24 %.

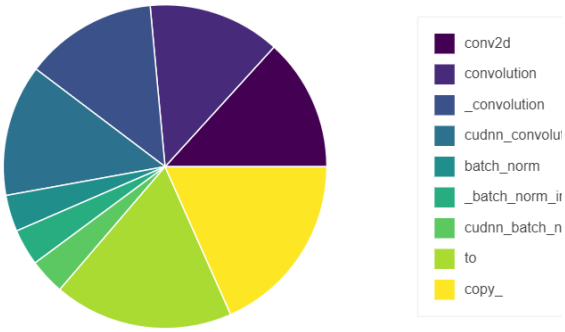
#	Percentage	Cumulative time in microseconds	CPU operator
0	24.46	4275788	copy_
1	23.94	4185809	to
2	12.9	2255234	conv2d
3	12.25	2142302	convolution
4	11.88	2076328	_convolution
5	7.63	1333065	cudnn_convolution
6	4.34	758263	batch_norm
7	2.6	454617	size



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 "copy_" with 18 %

#	Percentage	Cumulative time in microse	GPU operator
0	18.34	4100818	copy_
1	17.93	4007985	to
2	13.24	2960319	conv2d
3	13.23	2958240	convolution
4	13.22	2956267	_convolution
5	13.15	2940138	cudnn_convolution
6	3.65	815321	batch_norm
7	3.64	813525	_batch_norm_impl_index
8	3.58	801270	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 1231 datapoints and was triggered 46 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.	46	1231	
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.	2	1231	
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	
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	1239	
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	1239	
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	377	
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	1230	gpu_m
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.	0	377	
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	1231	

Analyzing the training loop

Step duration analysis

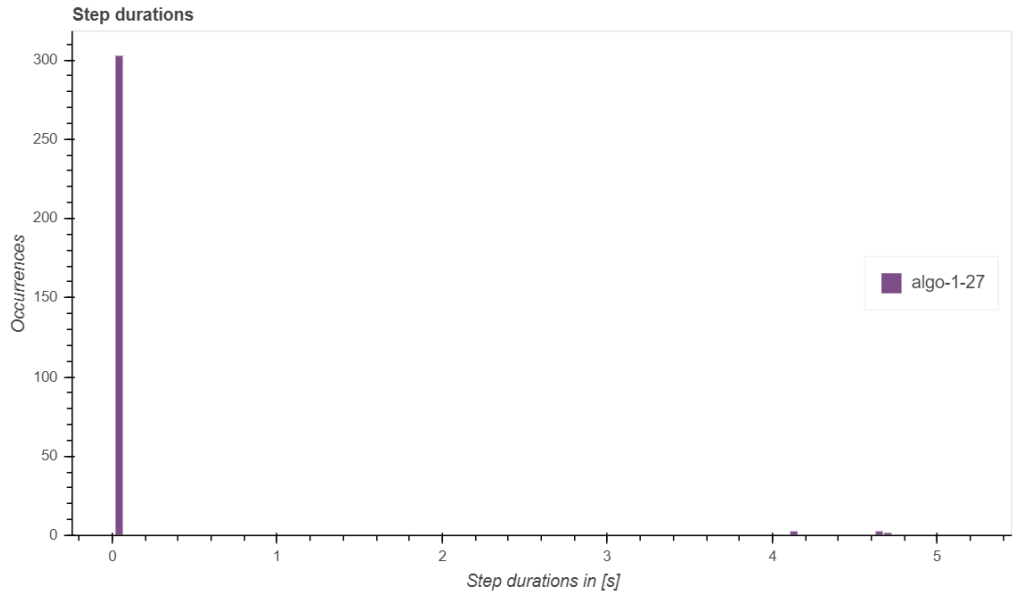
The StepOutlier rule measures step durations and checks for outliers. The rule returns True if duration is larger than 3 times the standard deviation. The rule also takes the parameter mode, that specifies whether steps from training or validation phase should be checked. In your processing job mode 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 377 datapoints and triggered 0 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:ModeKe average step duration on node algo-1-27 was 0.3s. The rule detected 2 outliers, where step duration was larger than 3 times the standard deviation of 1.89s

	mean	max	p99	p95	p50	min
Step Durations in [s]	0.30	21.85	4.71	0.03	0.02	0.02

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 o labels in the legend.



(http

GPU utilization analysis

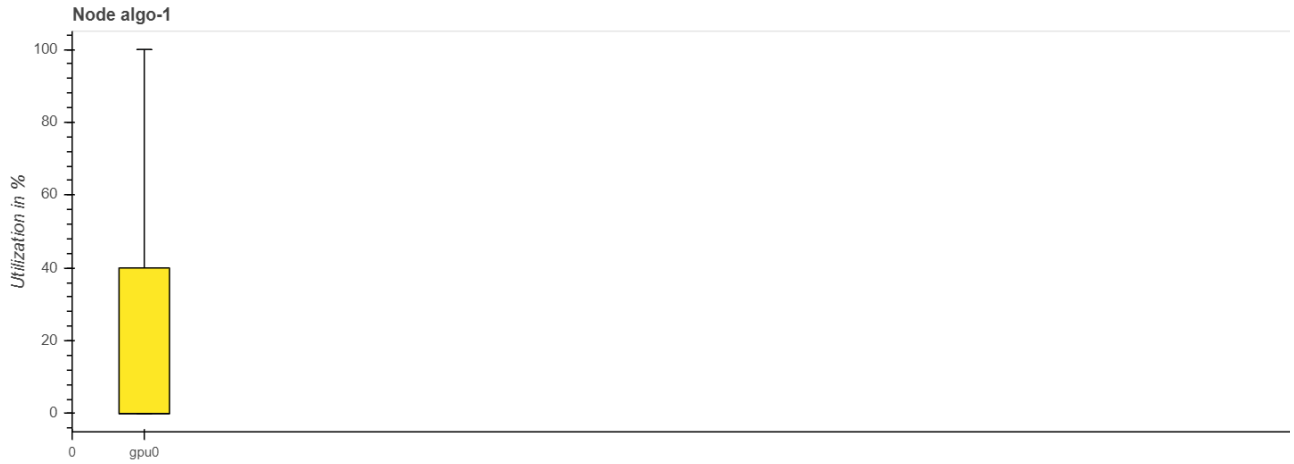
Usage per GPU

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 2 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 1231 datapoints and triggered 2 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 02/03/2023 at 05:55:00. 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:

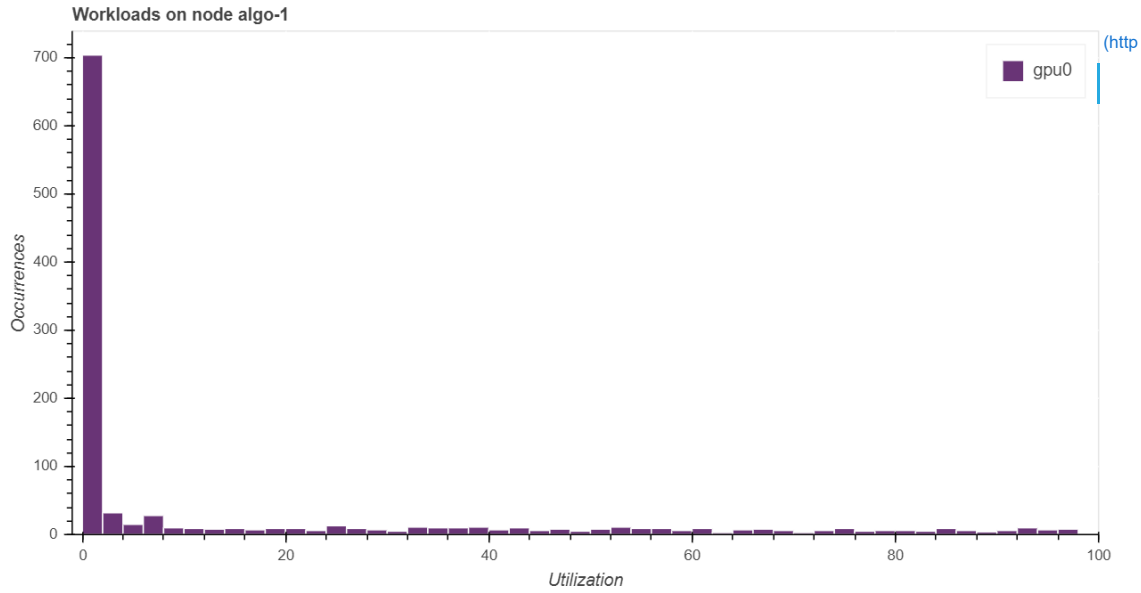
The max utilization of gpu0 on node algo-1 was 100.0% and the 5th percentile was only 0.0% The difference between 5th percentile 0.0% and 95th percentile 100.0% is quite significant, which means that utilization on gpu0 is fluctuating quite a lot.



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.



Dataloading analysis

The number of dataloader workers can greatly affect the overall performance of your training job. The rule analyzed the number of dataloading processes that have been running in parallel on the training instance and compares it against the total number of cores. The rule checked if the number of processes is smaller than 70% or larger than 200% the total number of cores. Having too few dataloader workers can slowdown data preprocessing and lead to GPU underutilization. Having too many dataloader workers may hurt the overall performance if you are running other compute intensive tasks on the CPU. The rule analysed 0 datapoints and triggered 0 times.

Batch size

In [35…
```
# Zipping the ProfilerReport inorder to export and upload it later for submission
import shutil
shutil.make_archive("./profiler_report", "zip", "ProfilerReport")
```

Out[35]:
```
'/root/deep_learning/profiler_report.zip'
```

Model Deploying

In [45…
```
# TODO: Deploy your model to an endpoint
predictor = estimator.deploy(initial_instance_count=1, instance_type="ml.m5.xlarge")
```

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

In [46…

```
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-1-804209765268/pytorch-training-230203-0456-004-2847225f/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")
```

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

In [65…

```
#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 = "./dogs_vs_cats/test/"
test_images = ["cats/cat.10261.jpg", "dogs/dog.10054.jpg", "cats/cat.10046.jpg", "dogs/dog.10126.jpg"]
test_images_expected_output = ['cat', 'dog', 'cat', 'dog' ]
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 dog breed 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 == 1:
            print(f"Response/Inference for the above image is : 'cat'")
            print("--------------------------------------------------------------------------")
        else:
            print(f"Response/Inference for the above image is : 'dog'")
            print("--------------------------------------------------------------------------")
```

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

```
Expected dog breed category no : cat
Response: [[20.534025192260742, 15.059690475463867, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'cat'
----------------------------------------------------------------------
Test image no: 2
Below is the image that we will be testing:
```

```
Expected dog breed category no : dog
Response: [[13.573927879333496, 18.657485961914062, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'dog'
----------------------------------------------------------------------
Test image no: 3
Below is the image that we will be testing:
```

```
Expected dog breed category no : cat
Response: [[24.11429214477539, 12.61474895477295, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'cat'
----------------------------------------------------------------------
Test image no: 4
Below is the image that we will be testing:
```

```
Expected dog breed category no : dog
Response: [[11.11713981628418, 18.990795135498047, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'dog'
----------------------------------------------------------------------
```

In [66…

```python
#  Using boto3
# Using the runtime boto3 client to test the deployed models endpoint
import os
import io
import boto3
import json
import base64
import PIL
# setting the  environment variables

ENDPOINT_NAME = "pytorch-inference-2023-02-03-08-06-56-724"
# We will be using the AWS's lightweight runtime solution to invoke an endpoint.
runtime= boto3.client('runtime.sagemaker')
test_dir = "./dogs_vs_cats/test/"
test_images = ["cats/cat.10261.jpg", "dogs/dog.10054.jpg", "cats/cat.10046.jpg", "dogs/dog.10126.jpg"]
test_images_expected_output = ['cat', 'dog', 'cat', 'dog' ]
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 dog breed 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 zero-indexed
        if predicted_category == 1:
            print(f"Response/Inference for the above image is : 'cat'")
            print("----------------------------------------------------------------------")
        else:
            print(f"Response/Inference for the above image is : 'dog'")
            print("----------------------------------------------------------------------")
```

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

```
Expected dog breed category no : cat
Response: [[20.534025192260742, 15.059690475463867, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'cat'
----------------------------------------------------------------------
Test image no: 2
Below is the image that we will be testing:
```

```
Expected dog breed category no : dog
Response: [[13.573927879333496, 18.657485961914062, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'dog'
----------------------------------------------------------------------
Test image no: 3
Below is the image that we will be testing:
```

```
Expected dog breed category no : cat
Response: [[24.11429214477539, 12.61474895477295, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'cat'
----------------------------------------------------------------------
Test image no: 4
Below is the image that we will be testing:
```

```
Expected dog breed category no : dog
Response: [[11.11713981628418, 18.990795135498047, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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 : 'dog'
----------------------------------------------------------------------
```

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

In [67…
```
print(predictor.endpoint_name)
endpoint_name = predictor.endpoint_name
```

```
pytorch-inference-2023-02-03-08-06-56-724
```

In [68…
```
# TODO: Remember to shutdown/delete your endpoint once your work is done
predictor.delete_endpoint()
```