Compile and Train a Hugging Face Transformer BERT Model with the SST Dataset using SageMaker Training Compiler

SageMaker Training Compiler Overview

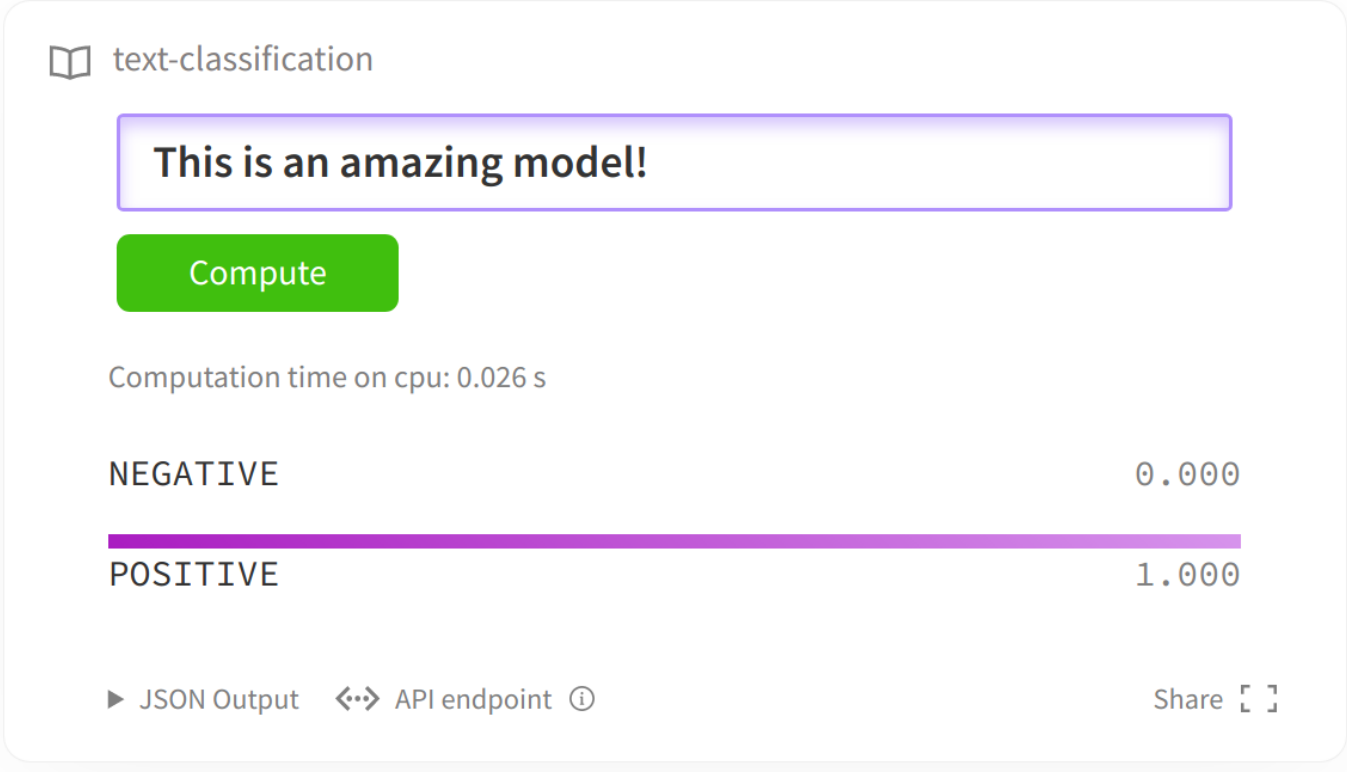
SageMaker Training Compiler is a capability of SageMaker that makes these hard-to-implement optimizations to reduce training time on GPU instances. The compiler optimizes DL models to accelerate training by more efficiently using SageMaker machine learning (ML) GPU instances. SageMaker Training Compiler is available at no additional charge within SageMaker and can help reduce total billable time as it accelerates training.

SageMaker Training Compiler is integrated into the AWS Deep Learning Containers (DLCs). Using the SageMaker Training Compiler enabled AWS DLCs, you can compile and optimize training jobs on GPU instances with minimal changes to your code. Bring your deep learning models to SageMaker and enable SageMaker Training Compiler to accelerate the speed of your training job on SageMaker ML instances for accelerated computing.

For more information, see SageMaker Training Compiler (https://docs.aws.amazon.com/sagemaker/latest/dg/training-compiler.html) in the *Amazon SageMaker Developer Guide*.

Introduction

This notebook is an end-to-end binary text classification example. In this demo, we use the Hugging Face's `transformers` and `datasets` libraries with SageMaker Training Compiler to compile and fine-tune a pre-trained transformer for binary text classification. In particular, the pre-trained model will be fine-tuned using the `Stanford Sentiment Treebank (SST)` dataset. To get started, you need to set up the environment with a few prerequisite steps, for permissions, configurations, and so on.



NOTE: You can run this demo in SageMaker Studio, SageMaker notebook instances, or your local machine with AWS CLI set up. If using SageMaker Studio or SageMaker notebook instances, make sure you choose one of the PyTorch-based kernels, `Python 3 (PyTorch x.y Python 3.x CPU Optimized)` or `conda_pytorch_p36` respectively.

NOTE: This notebook uses two `ml.p3.2xlarge` instances that have single GPU. If you don't have enough quota, see Request a service quota increase for SageMaker resources (https://docs.aws.amazon.com/sagemaker/latest/dg/regions-quotas.html#service-limit-increase-request-procedure).

Prepare SageMaker Environment and Permissions

Installation

This example notebook requires the **SageMaker Python SDK v2.108.0** and **transformers v4.21**.

In [1]:
```
!pip install "sagemaker>=2.108.0" botocore boto3 awscli s3fs typing
-extensions "torch==1.11.0" --upgrade
```

```
Looking in indexes: https://pypi.org/simple, https://pip.repos.neuro
n.amazonaws.com
Requirement already satisfied: sagemaker>=2.108.0 in /home/ec2-user/
anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (2.109.0)
Requirement already satisfied: botocore in /home/ec2-user/anaconda3/
envs/pytorch_p38/lib/python3.8/site-packages (1.27.72)
Requirement already satisfied: boto3 in /home/ec2-user/anaconda3/env
s/pytorch_p38/lib/python3.8/site-packages (1.24.72)
Requirement already satisfied: awscli in /home/ec2-user/anaconda3/en
vs/pytorch_p38/lib/python3.8/site-packages (1.25.73)
Requirement already satisfied: s3fs in /home/ec2-user/anaconda3/envs
/pytorch_p38/lib/python3.8/site-packages (0.4.2)
Collecting s3fs
  Using cached s3fs-2022.8.2-py3-none-any.whl (27 kB)
Requirement already satisfied: typing-extensions in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (4.3.0)
Requirement already satisfied: torch==1.11.0 in /home/ec2-user/anaco
nda3/envs/pytorch_p38/lib/python3.8/site-packages (1.11.0)
Requirement already satisfied: google-pasta in /home/ec2-user/anacon
da3/envs/pytorch_p38/lib/python3.8/site-packages (from sagemaker>=2.
108.0) (0.2.0)
Requirement already satisfied: pathos in /home/ec2-user/anaconda3/en
vs/pytorch_p38/lib/python3.8/site-packages (from sagemaker>=2.108.0)
(0.2.8)
Requirement already satisfied: protobuf<4.0,>=3.1 in /home/ec2-user/
anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from sagemak
er>=2.108.0) (3.20.1)
Requirement already satisfied: numpy<2.0,>=1.9.0 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from sagemake
r>=2.108.0) (1.21.2)
Requirement already satisfied: smdebug-rulesconfig==1.0.1 in /home/e
c2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from
sagemaker>=2.108.0) (1.0.1)
Requirement already satisfied: attrs<22,>=20.3.0 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from sagemake
r>=2.108.0) (21.2.0)
Requirement already satisfied: packaging>=20.0 in /home/ec2-user/ana
conda3/envs/pytorch_p38/lib/python3.8/site-packages (from sagemaker>
=2.108.0) (21.3)
Requirement already satisfied: protobuf3-to-dict<1.0,>=0.1.5 in /hom
e/ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (f
rom sagemaker>=2.108.0) (0.1.5)
Requirement already satisfied: pandas in /home/ec2-user/anaconda3/en
vs/pytorch_p38/lib/python3.8/site-packages (from sagemaker>=2.108.0)
(1.3.4)
Requirement already satisfied: importlib-metadata<5.0,>=1.4.0 in /ho
me/ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages
(from sagemaker>=2.108.0) (4.8.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /home/ec2-us
er/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from boto
core) (1.26.8)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /home/
ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (fro
m botocore) (2.8.2)
```

```
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /home/ec2-u
ser/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from bot
ocore) (0.10.0)
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in /home/ec2
-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from b
oto3) (0.6.0)
Requirement already satisfied: docutils<0.17,>=0.10 in /home/ec2-use
r/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from awscl
i) (0.15.2)
Requirement already satisfied: rsa<4.8,>=3.1.2 in /home/ec2-user/ana
conda3/envs/pytorch_p38/lib/python3.8/site-packages (from awscli)
(4.7.2)
Requirement already satisfied: colorama<0.4.5,>=0.2.5 in /home/ec2-u
ser/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from aws
cli) (0.4.3)
Requirement already satisfied: PyYAML<5.5,>=3.10 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from awscli)
(5.4.1)
Collecting fsspec==2022.8.2
  Using cached fsspec-2022.8.2-py3-none-any.whl (140 kB)
Collecting aiobotocore~=2.4.0
  Using cached aiobotocore-2.4.0-py3-none-any.whl (65 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /home/e
c2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from
s3fs) (3.8.1)
INFO: pip is looking at multiple versions of typing-extensions to de
termine which version is compatible with other requirements. This co
uld take a while.
Collecting typing-extensions
  Using cached typing_extensions-4.3.0-py3-none-any.whl (25 kB)
INFO: pip is looking at multiple versions of fsspec to determine whi
ch version is compatible with other requirements. This could take a
while.
INFO: pip is looking at multiple versions of <Python from Requires-P
ython> to determine which version is compatible with other requireme
nts. This could take a while.
INFO: pip is looking at multiple versions of s3fs to determine which
version is compatible with other requirements. This could take a whi
le.
Collecting s3fs
  Using cached s3fs-2022.8.1-py3-none-any.whl (27 kB)
Collecting fsspec==2022.8.1
  Using cached fsspec-2022.8.1-py3-none-any.whl (140 kB)
Collecting s3fs
  Using cached s3fs-2022.8.0-py3-none-any.whl (27 kB)
Collecting fsspec==2022.8.0
  Using cached fsspec-2022.8.0-py3-none-any.whl (140 kB)
Collecting s3fs
  Using cached s3fs-2022.7.1-py3-none-any.whl (27 kB)
Collecting aiobotocore~=2.3.4
  Using cached aiobotocore-2.3.4-py3-none-any.whl (64 kB)
Collecting fsspec==2022.7.1
  Using cached fsspec-2022.7.1-py3-none-any.whl (141 kB)
Collecting s3fs
```

```
  Using cached s3fs-2022.7.0-py3-none-any.whl (27 kB)
Collecting fsspec==2022.7.0
  Using cached fsspec-2022.7.0-py3-none-any.whl (141 kB)
Collecting s3fs
  Using cached s3fs-2022.5.0-py3-none-any.whl (27 kB)
Collecting fsspec==2022.5.0
  Using cached fsspec-2022.5.0-py3-none-any.whl (140 kB)
Requirement already satisfied: aiobotocore~=2.3.0 in /home/ec2-user/
anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from s3fs)
(2.3.3)
Collecting aiobotocore~=2.3.0
  Using cached aiobotocore-2.3.2.tar.gz (104 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-2.3.1.tar.gz (65 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-2.3.0.tar.gz (65 kB)
  Preparing metadata (setup.py) ... done
Collecting s3fs
  Using cached s3fs-2022.3.0-py3-none-any.whl (26 kB)
Collecting aiobotocore~=2.2.0
  Using cached aiobotocore-2.2.0.tar.gz (59 kB)
  Preparing metadata (setup.py) ... done
Collecting fsspec==2022.3.0
  Using cached fsspec-2022.3.0-py3-none-any.whl (136 kB)
Collecting s3fs
  Using cached s3fs-2022.2.0-py3-none-any.whl (26 kB)
Collecting fsspec==2022.02.0
  Using cached fsspec-2022.2.0-py3-none-any.whl (134 kB)
Collecting aiobotocore~=2.1.0
  Using cached aiobotocore-2.1.2.tar.gz (58 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-2.1.1.tar.gz (57 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-2.1.0.tar.gz (54 kB)
  Preparing metadata (setup.py) ... done
INFO: pip is looking at multiple versions of fsspec to determine whi
ch version is compatible with other requirements. This could take a
while.
INFO: pip is looking at multiple versions of <Python from Requires-P
ython> to determine which version is compatible with other requireme
nts. This could take a while.
INFO: pip is looking at multiple versions of s3fs to determine which
version is compatible with other requirements. This could take a whi
le.
Collecting s3fs
  Using cached s3fs-2022.1.0-py3-none-any.whl (25 kB)
Collecting fsspec==2022.01.0
  Using cached fsspec-2022.1.0-py3-none-any.whl (133 kB)
Collecting s3fs
  Using cached s3fs-2021.11.1-py3-none-any.whl (25 kB)
Requirement already satisfied: fsspec==2021.11.1 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from s3fs) (2
021.11.1)
Collecting aiobotocore~=2.0.1
```

```
  Using cached aiobotocore-2.0.1-py3-none-any.whl
Requirement already satisfied: aioitertools>=0.5.1 in /home/ec2-user
/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiobot
ocore~=2.0.1->s3fs) (0.8.0)
Requirement already satisfied: wrapt>=1.10.10 in /home/ec2-user/anac
onda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiobotocore
~=2.0.1->s3fs) (1.13.3)
Collecting fsspec==2021.11.1
  Using cached fsspec-2021.11.1-py3-none-any.whl (132 kB)
Collecting s3fs
  Using cached s3fs-2021.11.0-py3-none-any.whl (25 kB)
Collecting aiobotocore~=1.4.1
  Using cached aiobotocore-1.4.2.tar.gz (52 kB)
  Preparing metadata (setup.py) ... done
Collecting fsspec==2021.11.0
  Using cached fsspec-2021.11.0-py3-none-any.whl (132 kB)
Collecting aiobotocore~=1.4.1
  Using cached aiobotocore-1.4.1.tar.gz (52 kB)
  Preparing metadata (setup.py) ... done
Collecting s3fs
  Using cached s3fs-2021.10.1-py3-none-any.whl (26 kB)
Collecting fsspec==2021.10.1
  Using cached fsspec-2021.10.1-py3-none-any.whl (125 kB)
INFO: This is taking longer than usual. You might need to provide th
e dependency resolver with stricter constraints to reduce runtime. S
ee https://pip.pypa.io/warnings/backtracking for guidance. If you wa
nt to abort this run, press Ctrl + C.
Collecting s3fs
  Using cached s3fs-2021.10.0-py3-none-any.whl (26 kB)
Collecting fsspec==2021.10.0
  Using cached fsspec-2021.10.0-py3-none-any.whl (125 kB)
INFO: This is taking longer than usual. You might need to provide th
e dependency resolver with stricter constraints to reduce runtime. S
ee https://pip.pypa.io/warnings/backtracking for guidance. If you wa
nt to abort this run, press Ctrl + C.
INFO: This is taking longer than usual. You might need to provide th
e dependency resolver with stricter constraints to reduce runtime. S
ee https://pip.pypa.io/warnings/backtracking for guidance. If you wa
nt to abort this run, press Ctrl + C.
Collecting s3fs
  Using cached s3fs-2021.9.0-py3-none-any.whl (26 kB)
Collecting fsspec==2021.09.0
  Using cached fsspec-2021.9.0-py3-none-any.whl (123 kB)
Collecting s3fs
  Using cached s3fs-2021.8.1-py3-none-any.whl (26 kB)
Collecting fsspec==2021.08.1
  Using cached fsspec-2021.8.1-py3-none-any.whl (119 kB)
Collecting aiobotocore~=1.4.0
  Using cached aiobotocore-1.4.0.tar.gz (51 kB)
  Preparing metadata (setup.py) ... done
Collecting s3fs
  Using cached s3fs-2021.8.0-py3-none-any.whl (26 kB)
Collecting fsspec==2021.07.0
  Using cached fsspec-2021.7.0-py3-none-any.whl (118 kB)
```

```
Collecting s3fs
  Using cached s3fs-2021.7.0-py3-none-any.whl (25 kB)
Collecting aiobotocore>=1.0.1
  Using cached aiobotocore-2.0.0.tar.gz (52 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.3.3.tar.gz (50 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.3.2.tar.gz (49 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.3.1.tar.gz (48 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.3.0.tar.gz (48 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.2.2.tar.gz (48 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.2.1.tar.gz (48 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.2.0.tar.gz (47 kB)
  Preparing metadata (setup.py) ... done
  Using cached aiobotocore-1.1.2-py3-none-any.whl (45 kB)
  Using cached aiobotocore-1.1.1-py3-none-any.whl (45 kB)
  Using cached aiobotocore-1.1.0-py3-none-any.whl (43 kB)
  Using cached aiobotocore-1.0.7-py3-none-any.whl (42 kB)
  Using cached aiobotocore-1.0.6-py3-none-any.whl (42 kB)
  Using cached aiobotocore-1.0.5-py3-none-any.whl (42 kB)
  Using cached aiobotocore-1.0.4-py3-none-any.whl (41 kB)
  Using cached aiobotocore-1.0.3-py3-none-any.whl (40 kB)
  Using cached aiobotocore-1.0.2-py3-none-any.whl (40 kB)
  Using cached aiobotocore-1.0.1-py3-none-any.whl (40 kB)
Collecting s3fs
  Using cached s3fs-2021.6.1-py3-none-any.whl (25 kB)
Collecting fsspec==2021.06.1
  Using cached fsspec-2021.6.1-py3-none-any.whl (115 kB)
Collecting s3fs
  Using cached s3fs-2021.6.0-py3-none-any.whl (24 kB)
Collecting fsspec==2021.06.0
  Using cached fsspec-2021.6.0-py3-none-any.whl (114 kB)
Collecting s3fs
  Using cached s3fs-2021.5.0-py3-none-any.whl (24 kB)
Collecting fsspec==2021.05.0
  Using cached fsspec-2021.5.0-py3-none-any.whl (111 kB)
Collecting s3fs
  Using cached s3fs-2021.4.0-py3-none-any.whl (23 kB)
Collecting fsspec==2021.04.0
  Using cached fsspec-2021.4.0-py3-none-any.whl (108 kB)
Collecting s3fs
  Using cached s3fs-0.6.0-py3-none-any.whl (23 kB)
  Using cached s3fs-0.5.2-py3-none-any.whl (22 kB)
  Using cached s3fs-0.5.1-py3-none-any.whl (21 kB)
  Using cached s3fs-0.5.0-py3-none-any.whl (21 kB)
Requirement already satisfied: zipp>=0.5 in /home/ec2-user/anaconda3
/envs/pytorch_p38/lib/python3.8/site-packages (from importlib-metada
ta<5.0,>=1.4.0->sagemaker>=2.108.0) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /home/ec2
```

```
-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from p
ackaging>=20.0->sagemaker>=2.108.0) (3.0.6)
Requirement already satisfied: six in /home/ec2-user/anaconda3/envs/
pytorch_p38/lib/python3.8/site-packages (from protobuf3-to-dict<1.0,
>=0.1.5->sagemaker>=2.108.0) (1.16.0)
Requirement already satisfied: pyasn1>=0.1.3 in /home/ec2-user/anaco
nda3/envs/pytorch_p38/lib/python3.8/site-packages (from rsa<4.8,>=3.
1.2->awscli) (0.4.8)
Requirement already satisfied: pytz>=2017.3 in /home/ec2-user/anacon
da3/envs/pytorch_p38/lib/python3.8/site-packages (from pandas->sagem
aker>=2.108.0) (2021.3)
Requirement already satisfied: pox>=0.3.0 in /home/ec2-user/anaconda
3/envs/pytorch_p38/lib/python3.8/site-packages (from pathos->sagemak
er>=2.108.0) (0.3.0)
Requirement already satisfied: dill>=0.3.4 in /home/ec2-user/anacond
a3/envs/pytorch_p38/lib/python3.8/site-packages (from pathos->sagema
ker>=2.108.0) (0.3.4)
Requirement already satisfied: ppft>=1.6.6.4 in /home/ec2-user/anaco
nda3/envs/pytorch_p38/lib/python3.8/site-packages (from pathos->sage
maker>=2.108.0) (1.6.6.4)
Requirement already satisfied: multiprocess>=0.70.12 in /home/ec2-us
er/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from path
os->sagemaker>=2.108.0) (0.70.12.2)
WARNING: You are using pip version 22.0.4; however, version 22.2.2 i
```

In [2]:
```
!pip install "transformers==4.21" datasets --upgrade
```

```
Looking in indexes: https://pypi.org/simple, https://pip.repos.neuro
n.amazonaws.com
Collecting transformers==4.21
  Using cached transformers-4.21.0-py3-none-any.whl (4.7 MB)
Requirement already satisfied: datasets in /home/ec2-user/anaconda3/
envs/pytorch_p38/lib/python3.8/site-packages (2.4.0)
Requirement already satisfied: numpy>=1.17 in /home/ec2-user/anacond
a3/envs/pytorch_p38/lib/python3.8/site-packages (from transformers==
4.21) (1.21.2)
Requirement already satisfied: packaging>=20.0 in /home/ec2-user/ana
conda3/envs/pytorch_p38/lib/python3.8/site-packages (from transforme
rs==4.21) (21.3)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in /home/
ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (fro
m transformers==4.21) (0.9.0)
Requirement already satisfied: tqdm>=4.27 in /home/ec2-user/anaconda
3/envs/pytorch_p38/lib/python3.8/site-packages (from transformers==
4.21) (4.62.3)
Requirement already satisfied: requests in /home/ec2-user/anaconda3/
envs/pytorch_p38/lib/python3.8/site-packages (from transformers==4.2
1) (2.26.0)
Requirement already satisfied: pyyaml>=5.1 in /home/ec2-user/anacond
a3/envs/pytorch_p38/lib/python3.8/site-packages (from transformers==
4.21) (5.4.1)
Requirement already satisfied: regex!=2019.12.17 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from transfor
mers==4.21) (2021.11.10)
Requirement already satisfied: tokenizers!=0.11.3,<0.13,>=0.11.1 in
/home/ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-package
s (from transformers==4.21) (0.12.1)
Requirement already satisfied: filelock in /home/ec2-user/anaconda3/
envs/pytorch_p38/lib/python3.8/site-packages (from transformers==4.2
1) (3.4.0)
Requirement already satisfied: fsspec[http]>=2021.11.1 in /home/ec2-
user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from da
tasets) (2021.11.1)
Requirement already satisfied: multiprocess in /home/ec2-user/anacon
da3/envs/pytorch_p38/lib/python3.8/site-packages (from datasets) (0.
70.12.2)
Requirement already satisfied: dill<0.3.6 in /home/ec2-user/anaconda
3/envs/pytorch_p38/lib/python3.8/site-packages (from datasets) (0.3.
4)
Requirement already satisfied: xxhash in /home/ec2-user/anaconda3/en
vs/pytorch_p38/lib/python3.8/site-packages (from datasets) (3.0.0)
Requirement already satisfied: aiohttp in /home/ec2-user/anaconda3/e
nvs/pytorch_p38/lib/python3.8/site-packages (from datasets) (3.8.1)
Requirement already satisfied: responses<0.19 in /home/ec2-user/anac
onda3/envs/pytorch_p38/lib/python3.8/site-packages (from datasets)
(0.18.0)
Requirement already satisfied: pandas in /home/ec2-user/anaconda3/en
vs/pytorch_p38/lib/python3.8/site-packages (from datasets) (1.3.4)
Requirement already satisfied: pyarrow>=6.0.0 in /home/ec2-user/anac
onda3/envs/pytorch_p38/lib/python3.8/site-packages (from datasets)
(7.0.0)
```

```
Requirement already satisfied: typing-extensions>=3.7.4.3 in /home/e
c2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from
huggingface-hub<1.0,>=0.1.0->transformers==4.21) (4.3.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /home/ec2
-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from p
ackaging>=20.0->transformers==4.21) (3.0.6)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/ec2-us
er/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from requ
ests->transformers==4.21) (1.26.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in /home/ec
2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from
requests->transformers==4.21) (2.0.7)
Requirement already satisfied: idna<4,>=2.5 in /home/ec2-user/anacon
da3/envs/pytorch_p38/lib/python3.8/site-packages (from requests->tra
nsformers==4.21) (3.1)
Requirement already satisfied: certifi>=2017.4.17 in /home/ec2-user/
anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from request
s->transformers==4.21) (2021.10.8)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /home/
ec2-user/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (fro
m aiohttp->datasets) (4.0.1)
Requirement already satisfied: yarl<2.0,>=1.0 in /home/ec2-user/anac
onda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiohttp->da
tasets) (1.7.2)
Requirement already satisfied: frozenlist>=1.1.1 in /home/ec2-user/a
naconda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiohttp-
>datasets) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in /home/ec2-user/anaco
nda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiohttp->dat
asets) (21.2.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /home/ec2-user
/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiohtt
p->datasets) (5.2.0)
Requirement already satisfied: aiosignal>=1.1.2 in /home/ec2-user/an
aconda3/envs/pytorch_p38/lib/python3.8/site-packages (from aiohttp->
datasets) (1.2.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/ec2-u
ser/anaconda3/envs/pytorch_p38/lib/python3.8/site-packages (from pan
das->datasets) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/ec2-user/anacon
da3/envs/pytorch_p38/lib/python3.8/site-packages (from pandas->datas
ets) (2021.3)
Requirement already satisfied: six>=1.5 in /home/ec2-user/anaconda3/
envs/pytorch_p38/lib/python3.8/site-packages (from python-dateutil>=
2.7.3->pandas->datasets) (1.16.0)
Installing collected packages: transformers
  Attempting uninstall: transformers
    Found existing installation: transformers 4.21.1
    Uninstalling transformers-4.21.1:
      Successfully uninstalled transformers-4.21.1
Successfully installed transformers-4.21.0
WARNING: You are using pip version 22.0.4; however, version 22.2.2 i
s available.
You should consider upgrading via the '/home/ec2-user/anaconda3/envs
```

In [3]:
```
import botocore
import boto3
import sagemaker
import transformers

print(f"sagemaker: {sagemaker.__version__}")
print(f"transformers: {transformers.__version__}")
```

```
sagemaker: 2.109.0
transformers: 4.21.0
```

Copy and run the following code if you need to upgrade "ipywidgets" for `datasets` library and restart kernel. This is only needed when preprocessing is done in the notebook.

```
%%capture
import IPython
!conda install -c conda-forge ipywidgets -y
# has to restart kernel for the updates to be applied
IPython.Application.instance().kernel.do_shutdown(True)
```

SageMaker environment

Note: If you are going to use SageMaker in a local environment. You need access to an IAM Role with the required permissions for SageMaker. To learn more, see SageMaker Roles (https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-roles.html).

In [4]:
```
import sagemaker

sess = sagemaker.Session()
# sagemaker session bucket -> used for uploading data, models and l
ogs
# sagemaker will automatically create this bucket if it not exists
sagemaker_session_bucket = None
if sagemaker_session_bucket is None and sess is not None:
    # set to default bucket if a bucket name is not given
    sagemaker_session_bucket = sess.default_bucket()

role = sagemaker.get_execution_role()
sess = sagemaker.Session(default_bucket=sagemaker_session_bucket)

print(f"sagemaker role arn: {role}")
print(f"sagemaker bucket: {sess.default_bucket()}")
print(f"sagemaker session region: {sess.boto_region_name}")
```

```
sagemaker role arn: arn:aws:iam::875423407011:role/AdminRole
sagemaker bucket: sagemaker-us-west-2-875423407011
sagemaker session region: us-west-2
```

Loading the SST dataset

When using the 🤗 Datasets library (https://github.com/huggingface/datasets), datasets can be downloaded directly with the following `datasets.load_dataset()` method:

```
from datasets import load_dataset
load_dataset('dataset_name')
```

If you'd like to try other training datasets later, you can simply use this method.

For this example notebook, we prepared the SST2 dataset (https://www.tensorflow.org/datasets/catalog/glue#gluesst2) in the public SageMaker sample S3 bucket. The following code cells show how you can directly load the dataset and convert to a `HuggingFace DatasetDict` .

Tokenization

In [5]:
```
from datasets import load_dataset
from transformers import AutoTokenizer
from datasets import Dataset

# tokenizer used in preprocessing
tokenizer_name = "bert-base-cased"

# dataset used
dataset_name = "sst"
```

In [6]:
```
import pandas as pd

# Read our data into pandas DataFrames to prepare it for training
# Initially our dataset will contain one column called 'line'

test_df = pd.read_csv(
    "https://sagemaker-sample-files.s3.amazonaws.com/datasets/text/
SST2/sst2.test",
    sep="delimiter",
    header=None,
    engine="python",
    names=["line"],
)
train_df = pd.read_csv(
    "https://sagemaker-sample-files.s3.amazonaws.com/datasets/text/
SST2/sst2.train",
    sep="delimiter",
    header=None,
    engine="python",
    names=["line"],
)
val_df = pd.read_csv(
    "https://sagemaker-sample-files.s3.amazonaws.com/datasets/text/
SST2/sst2.val",
    sep="delimiter",
    header=None,
    engine="python",
    names=["line"],
)
```

In [7]:
```
# Split each row into two columns one for the label and one for the
text

test_df[["label", "text"]] = test_df["line"].str.split(" ", 1, expa
nd=True)
train_df[["label", "text"]] = train_df["line"].str.split(" ", 1, ex
pand=True)
val_df[["label", "text"]] = val_df["line"].str.split(" ", 1, expand
=True)
```

In [8]:
```
# Drop the line column as it is no longer needed since we split thi
s column above

test_df.drop("line", axis=1, inplace=True)
train_df.drop("line", axis=1, inplace=True)
val_df.drop("line", axis=1, inplace=True)
```

In [9]:
```
# Convert the label into numeric instead of object type to prepare
binary labels for training
# After conversion the labels will be 8 bit integers

test_df["label"] = pd.to_numeric(test_df["label"], downcast="intege
r")
train_df["label"] = pd.to_numeric(train_df["label"], downcast="inte
ger")
val_df["label"] = pd.to_numeric(val_df["label"], downcast="intege
r")
```

In [10]:
```
# Combine test and val data into one df

test_df = pd.concat([test_df, val_df], ignore_index=True)
```

In [11]:
```
train_dataset = Dataset.from_pandas(train_df)
test_dataset = Dataset.from_pandas(test_df)
```

In [12]:
```
# Download tokenizer
# The tokenizer is loaded from the AutoTokenizer class and we use t
he from_pretrained method
# This allows us to instatiate a tokenizer based on a pretrained mo
del
tokenizer = AutoTokenizer.from_pretrained(tokenizer_name)

# Tokenizer helper function
# This function specifies the input should be tokenized by padding
to the max_length which is 512
# Anything beyond this length will be truncated
# This function will convert the 'text' column to a set of numeric
input ids that can be used for
# model training
# For more information on tokenization see the Hugging Face documen
tation: https://huggingface.co/transformers/preprocessing.html
def tokenize(batch):
    return tokenizer(batch["text"], padding="max_length", truncatio
n=True)


# Tokenize dataset in batches
# See Hugging Face documentation for more info on the map method: h
ttps://huggingface.co/docs/datasets/package_reference/main_classes.
html#datasets.Dataset.map
train_dataset = train_dataset.map(tokenize, batched=True)
test_dataset = test_dataset.map(tokenize, batched=True)
```

In [13]:
```
train_dataset = train_dataset.rename_column("label", "labels")
train_dataset.set_format("torch", columns=["input_ids", "attention_
mask", "labels"])
test_dataset = test_dataset.rename_column("label", "labels")
test_dataset.set_format("torch", columns=["input_ids", "attention_m
ask", "labels"])
```

Uploading data to `sagemaker_session_bucket`

After we processed the `datasets` we are going to use the new `FileSystem` integration (https://huggingface.co/docs/datasets/filesystems.html) to upload our dataset to S3.

In [14]:
```
import botocore
from datasets.filesystems import S3FileSystem

s3 = S3FileSystem()

# s3 key prefix for setting up the data channel for the current Sag
eMaker session
s3_prefix = "samples/datasets/sst"

# save train_dataset to s3
training_input_path = f"s3://{sess.default_bucket()}/{s3_prefix}/tr
ain"
train_dataset.save_to_disk(training_input_path, fs=s3)

# save test_dataset to s3
test_input_path = f"s3://{sess.default_bucket()}/{s3_prefix}/test"
test_dataset.save_to_disk(test_input_path, fs=s3)
```

SageMaker Training Job

To create a SageMaker training job, we use a `HuggingFace/PyTorch` estimator. Using the estimator, you can define which fine-tuning script should SageMaker use through `entry_point`, which `instance_type` to use for training, which `hyperparameters` to pass, and so on.

When a SageMaker training job starts, SageMaker takes care of starting and managing all the required machine learning instances, picks up the `HuggingFace` Deep Learning Container, uploads your training script, and downloads the data from `sagemaker_session_bucket` into the container at `/opt/ml/input/data`.

In the following section, you learn how to set up two versions of the SageMaker `HuggingFace/PyTorch` estimator, a native one without the compiler and an optimized one with the compiler.

In [15]:
```
!pygmentize ./scripts/train.py
```

```
huggingface/tokenizers: The current process just got forked, after p
arallelism has already been used. Disabling parallelism to avoid dea
dlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLE
LISM=(true | false)
from transformers import AutoModelForSequenceClassification, Traine
r, TrainingArguments, AutoTokenizer, TrainerCallback
from sklearn.metrics import accuracy_score, precision_recall_fscore_
support
from datasets import load_from_disk
import random
import logging
import sys
import argparse
import os
import torch
import numpy as np


if __name__ == "__main__":

    parser = argparse.ArgumentParser()

    # hyperparameters sent by the client are passed as command-line
arguments to the script.
    parser.add_argument("--epochs", type=int, default=50)
    parser.add_argument("--train_batch_size", type=int, default=32)
    parser.add_argument("--eval_batch_size", type=int, default=64)
    parser.add_argument("--warmup_steps", type=int, default=500)
    parser.add_argument("--model_name", type=str)
    parser.add_argument("--learning_rate", type=str, default=5e-5)

    # Data, model, and output directories
    parser.add_argument("--output_data_dir", type=str, default=os.en
viron["SM_OUTPUT_DATA_DIR"])
    parser.add_argument("--model_dir", type=str, default=os.environ
["SM_MODEL_DIR"])
    parser.add_argument("--n_gpus", type=str, default=os.environ["SM
_NUM_GPUS"])
    parser.add_argument("--training_dir", type=str, default=os.envir
on["SM_CHANNEL_TRAIN"])
    parser.add_argument("--test_dir", type=str, default=os.environ["
SM_CHANNEL_TEST"])

    args, _ = parser.parse_known_args()

    args.learning_rate = float(args.learning_rate)

    os.environ['GPU_NUM_DEVICES']=args.n_gpus

    # Set up logging
    logger = logging.getLogger(__name__)
```

```
    logging.basicConfig(
        level=logging.getLevelName("INFO"),
        handlers=[logging.StreamHandler(sys.stdout)],
        format="%(asctime)s - %(name)s - %(levelname)s - %(message)
s",
    )

    # load datasets
    train_dataset = load_from_disk(args.training_dir)
    test_dataset = load_from_disk(args.test_dir)

    # compute metrics function for binary classification
    def compute_metrics(pred):
        labels = pred.label_ids
        preds = pred.predictions.argmax(-1)
        precision, recall, f1, _ = precision_recall_fscore_support(l
abels, preds, average="binary")
        acc = accuracy_score(labels, preds)
        return {"accuracy": acc, "f1": f1, "precision": precision, "
recall": recall}

    # download model from model hub
    model = AutoModelForSequenceClassification.from_pretrained(args.
model_name)
    tokenizer = AutoTokenizer.from_pretrained(args.model_name)

    # define training args
    training_args = TrainingArguments(
        output_dir=args.model_dir,
        num_train_epochs=args.epochs,
        per_device_train_batch_size=args.train_batch_size,
        per_device_eval_batch_size=args.eval_batch_size,
        warmup_steps=args.warmup_steps,
        logging_dir=f"{args.output_data_dir}/logs",
        learning_rate=args.learning_rate,
        fp16=True,
        dataloader_drop_last=True,
        disable_tqdm=True,
        evaluation_strategy="no",
        save_strategy="no",
        save_total_limit=1,
        logging_strategy="epoch",
    )

    # create Trainer instance
    trainer = Trainer(
        model=model,
        args=training_args,
        compute_metrics=compute_metrics,
        train_dataset=train_dataset,
        eval_dataset=test_dataset,
        tokenizer=tokenizer,
    )
```

```
    # train model
    trainer.train()

    # evaluate model
    eval_result = trainer.evaluate(eval_dataset=test_dataset)

    # writes eval result to file which can be accessed later in s3 o
uput
    with open(os.path.join(args.output_data_dir, "eval_results.tx
t"), "w") as writer:
        print(f"***** Eval results *****")
        for key, value in sorted(eval_result.items()):
            writer.write(f"{key} = {value}\n")

    # Saves the model to s3
```

Training a PyTorch Trainer without compiling

In [16]:
```
from sagemaker.pytorch import PyTorch

hyperparameters = {"epochs": 5, "train_batch_size": 16, "model_nam
e": "bert-base-cased"}

# Scale the learning rate by batch size, as original LR was using b
atch size of 32
hyperparameters["learning_rate"] = float("5e-5") / 32 * hyperparame
ters["train_batch_size"]

# Scale the volume size by number of epochs
volume_size = 60 + 2 * hyperparameters["epochs"]
```

In [17]:
```
# By setting the hyperparameters in the PyTorch Estimator below
# and using the AutoModelForSequenceClassification class in the tra
in.py script
# we can fine-tune the bert-base-cased pretrained Transformer for s
equence classification

native_estimator = PyTorch(
    entry_point="train.py",
    source_dir="./scripts",
    instance_type="ml.p3.2xlarge",
    instance_count=1,
    role=role,
    py_version="py38",
    base_job_name="native-sst-bert-base-cased-p3-2x-pytorch-190",
    volume_size=volume_size,
    transformers_version="4.21.1",
    framework_version="1.11.0",
    hyperparameters=hyperparameters,
    disable_profiler=True,
    debugger_hook_config=False,
)

# starting the train job with our uploaded datasets as input
native_estimator.fit({"train": training_input_path, "test": test_in
put_path}, wait=False)

# The name of the training job. You might need to note this down in
case your kernel crashes.
native_estimator.latest_training_job.name
```

Out[17]:
```
'native-sst-bert-base-cased-p3-2x-pytorc-2022-09-14-16-51-08-096'
```

Training a PyTorch Trainer with SageMaker Training Compiler

Compilation through Training Compiler changes the memory footprint of the model. Most commonly, this manifests as a reduction in memory utilization and a consequent increase in the largest batch size that can fit on the GPU. Note that if you want to change the batch size, you must adjust the learning rate appropriately.

Note: We recommend you to turn the SageMaker Debugger's profiling and debugging tools off when you use compilation to avoid additional overheads.

We use the tested batch size that's provided at Tested Models (https://docs.aws.amazon.com/sagemaker/latest/dg/training-compiler-support.html#training-compiler-tested-models) in the *SageMaker Training Compiler Developer Guide*.

In [18]:
```
from sagemaker.huggingface import HuggingFace, TrainingCompilerConf
ig

hyperparameters = {
    "epochs": 5,
    "train_batch_size": 24,
    "model_name": "bert-base-cased",
}

# Scale the learning rate by batch size, as original LR was using b
atch size of 32
hyperparameters["learning_rate"] = float("5e-5") / 32 * hyperparame
ters["train_batch_size"]

# Scale the volume size by number of epochs
volume_size = 60 + 2 * hyperparameters["epochs"]
```

In [19]:
```
# By setting the hyperparameters in the PyTorch Estimator below
# and using the AutoModelForSequenceClassification class in the tra
in.py script
# the bert-base-cased pretrained Transformer is fine-tuned for sequ
ence classification

# Importantly, the TrainingCompilerConfig() is passed below to enab
le the SageMaker Training Compiler

sm_training_compiler_estimator = HuggingFace(
    entry_point="train.py",
    source_dir="./scripts",
    instance_type="ml.p3.2xlarge",
    instance_count=1,
    role=role,
    py_version="py38",
    base_job_name="sm-compiled-sst-bert-base-cased-p3-2x-pytorch-19
0",
    volume_size=volume_size,
    transformers_version="4.21.1",
    pytorch_version="1.11.0",
    compiler_config=TrainingCompilerConfig(),
    hyperparameters=hyperparameters,
    disable_profiler=True,
    debugger_hook_config=False,
)

# starting the train job with our uploaded datasets as input
sm_training_compiler_estimator.fit(
    {"train": training_input_path, "test": test_input_path}, wait=F
alse
)

# The name of the training job. You might need to note this down in
case your kernel crashes.
sm_training_compiler_estimator.latest_training_job.name
```

Out[19]:
```
'sm-compiled-sst-bert-base-cased-p3-2x-p-2022-09-14-16-51-08-902'
```

Wait for the training jobs to complete

In [20]:
```
waiter = native_estimator.sagemaker_session.sagemaker_client.get_waiter(
    "training_job_completed_or_stopped"
)
waiter.wait(TrainingJobName=native_estimator.latest_training_job.name)
waiter = sm_training_compiler_estimator.sagemaker_session.sagemaker_client.get_waiter(
    "training_job_completed_or_stopped"
)
waiter.wait(TrainingJobName=sm_training_compiler_estimator.latest_training_job.name)
```

Analysis and Results

Load information and logs of the training job *without* SageMaker Training Compiler

In [21]:
```
# container image used for native training job
print(f"container image used for training job: \n{native_estimator.image_uri}\n")

# s3 uri where the native trained model is located
print(f"s3 uri where the trained model is located: \n{native_estimator.model_data}\n")

# latest training job name for this estimator
print(
    f"latest training job name for this estimator: \n{native_estimator.latest_training_job.name}\n"
)
```

```
container image used for training job:
None

s3 uri where the trained model is located:
s3://sagemaker-us-west-2-875423407011/native-sst-bert-base-cased-p3-2x-pytorc-2022-09-14-16-51-08-096/output/model.tar.gz

latest training job name for this estimator:
native-sst-bert-base-cased-p3-2x-pytorc-2022-09-14-16-51-08-096
```

In [22]:
```
%%capture native

# access the logs of the native training job
native_estimator.sagemaker_session.logs_for_job(native_estimator.latest_training_job.name)
```

Note: If the estimator object is no longer available due to a kernel break or refresh, you need to directly use the training job name and manually attach the training job to a new native estimator. For example:

```
native_estimator = HuggingFace.attach("your_native_training_job_name")
```

Load information and logs of the training job *with* SageMaker Training Compiler

In [23]:
```
# container image used for optimized training job
print(f"container image used for training job: \n{sm_training_compi
ler_estimator.image_uri}\n")

# s3 uri where the optimized trained model is located
print(f"s3 uri where the trained model is located: \n{sm_training_c
ompiler_estimator.model_data}\n")

# latest training job name for this estimator
print(
    f"latest training job name for this estimator: \n{sm_training_c
ompiler_estimator.latest_training_job.name}\n"
)
```

```
container image used for training job:
None

s3 uri where the trained model is located:
s3://sagemaker-us-west-2-875423407011/sm-compiled-sst-bert-base-case
d-p3-2x-p-2022-09-14-16-51-08-902/output/model.tar.gz

latest training job name for this estimator:
sm-compiled-sst-bert-base-cased-p3-2x-p-2022-09-14-16-51-08-902
```

In [24]:
```
%%capture optimized

# access the logs of the optimized training job
sm_training_compiler_estimator.sagemaker_session.logs_for_job(
    sm_training_compiler_estimator.latest_training_job.name
)
```

Note: If the estimator object is no longer available due to a kernel break or refresh, you need to directly use the training job name and manually attach the training job to a new HuggingFace estimator. You may be able to retrieve the training job name from above where it was printed out, but the name can also be retrieved using the SageMaker Service Console, the SageMaker SDK, or the AWS CLI (https://docs.aws.amazon.com/cli/latest/reference/sagemaker/list-training-jobs.html (https://docs.aws.amazon.com/cli/latest/reference/sagemaker/list-training-jobs.html))

For example:

```
sm_training_compiler_estimator = HuggingFace.attach("your_compiled_hugging
face_training_job_name")
```

Create helper functions for analysis

In [25]:
```
from ast import literal_eval
from collections import defaultdict
from matplotlib import pyplot as plt

# Intermediary function for processing each line of stdout captured
# Remove leading and trailing whitespace and append data in curly b
races
# to final list
def _summarize(captured):
    final = []
    for line in captured.stdout.split("\n"):
        cleaned = line.strip()
        if "{" in cleaned and "}" in cleaned:
            final.append(cleaned[cleaned.index("{") : cleaned.index
("}") + 1])
    return final


# Check input with literal_eval
# https://docs.python.org/3/library/ast.html
def make_sense(string):
    try:
        return literal_eval(string)
    except:
        pass


# Parse the stdout and organize by train, evaluation, and summary d
ata
def summarize(summary):
    final = {"train": [], "eval": [], "summary": {}}
    for line in summary:
        interpretation = make_sense(line)
        if interpretation:
            if "loss" in interpretation:
                final["train"].append(interpretation)
            elif "eval_loss" in interpretation:
                final["eval"].append(interpretation)
            elif "train_runtime" in interpretation:
                final["summary"].update(interpretation)
    return final
```

Plot and compare throughput of compiled training and native training

In [26]:
```
native_summary = summarize(_summarize(native))
native_throughput = native_summary["summary"]["train_samples_per_second"]

optimized_summary = summarize(_summarize(optimized))
optimized_throughput = optimized_summary["summary"]["train_samples_per_second"]

avg_speedup = f"{round((optimized_throughput/native_throughput-1)*100)}%"
```

In [27]:
```
native_summary["summary"]
```

Out[27]:
```
{'train_runtime': 4816.0007,
 'train_samples_per_second': 69.922,
 'train_steps_per_second': 4.37,
 'train_loss': 0.11401634877921567,
 'epoch': 5.0}
```

Example output for native training job

{'train_runtime': 5258.1061, 'train_samples_per_second': 64.043, 'train_steps_per_second': 4.574, 'train_loss': 0.12156725528582218, 'epoch': 5.0}

In [28]:
```
optimized_summary["summary"]
```

Out[28]:
```
{'train_runtime': 3778.5479,
 'train_samples_per_second': 89.12,
 'train_steps_per_second': 3.713,
 'train_loss': 0.11231994411389656,
 'epoch': 5.0}
```

Example output for SageMaker Training Compiler training job

{'train_runtime': 3742.9028, 'train_samples_per_second': 89.969, 'train_steps_per_second': 3.748, 'train_loss': 0.11423833388901572, 'epoch': 5.0}

Training Throughput Plot

The following script creates a plot that compares the throughput (number_of_samples/second) of the two training jobs with and without SageMaker Training Compiler.

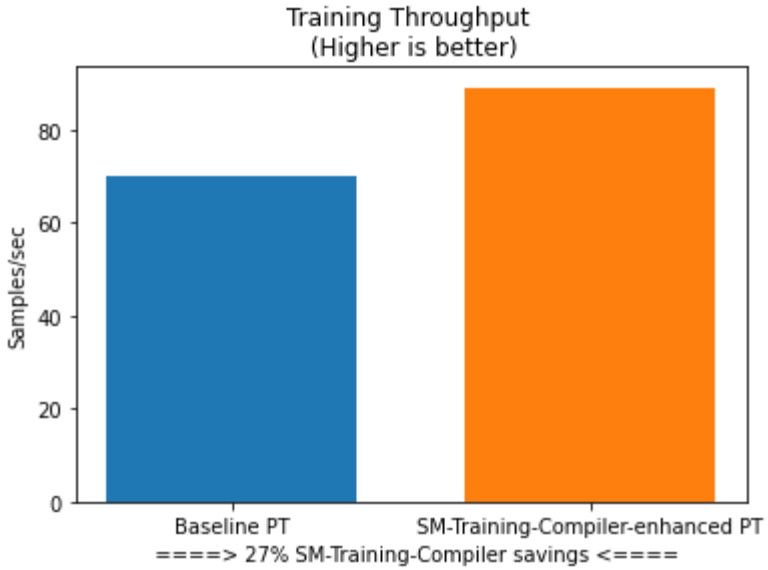
In [29]:
```
%matplotlib inline

plt.title("Training Throughput \n (Higher is better)")
plt.ylabel("Samples/sec")

plt.bar(x=[1], height=native_throughput, label="Baseline PT", width
=0.35)
plt.bar(x=[1.5], height=optimized_throughput, label="SM-Training-Co
mpiler-enhanced PT", width=0.35)

plt.xlabel("  ====> {} SM-Training-Compiler savings <====".format(a
vg_speedup))
plt.xticks(ticks=[1, 1.5], labels=["Baseline PT", "SM-Training-Comp
iler-enhanced PT"])
```

Out[29]:
```
([<matplotlib.axis.XTick at 0x7f132a865190>,
  <matplotlib.axis.XTick at 0x7f132a865160>],
 [Text(1.0, 0, 'Baseline PT'),
  Text(1.5, 0, 'SM-Training-Compiler-enhanced PT')])
```

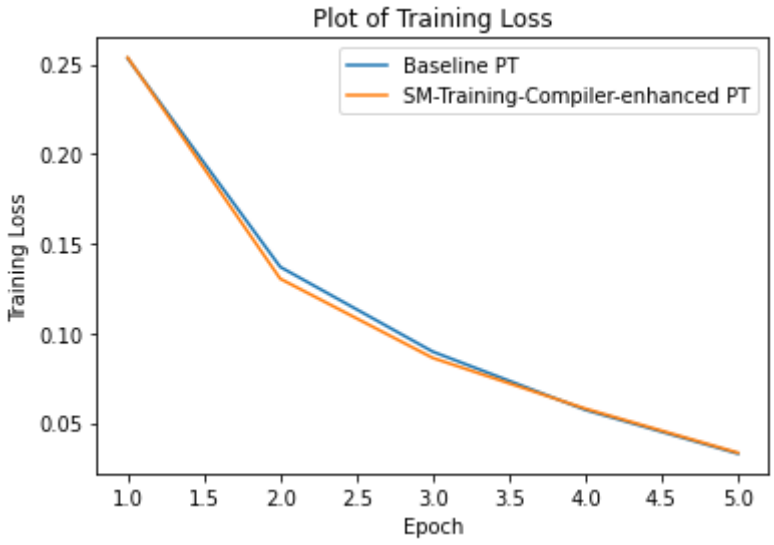


Convergence of Training Loss

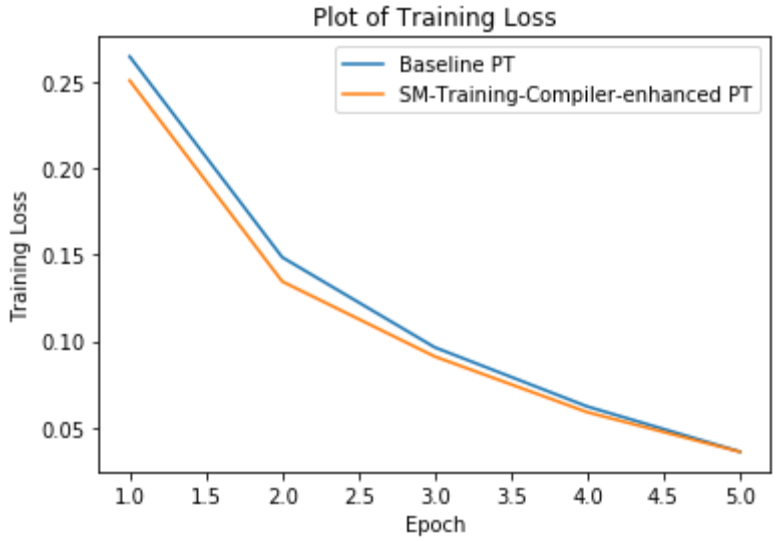
The following script creates a plot that compares the loss function of the two training jobs with and without SageMaker Training Compiler.

In [30]:
```
vanilla_loss = [i["loss"] for i in native_summary["train"]]
vanilla_epochs = [i["epoch"] for i in native_summary["train"]]
optimized_loss = [i["loss"] for i in optimized_summary["train"]]
optimized_epochs = [i["epoch"] for i in optimized_summary["train"]]

plt.title("Plot of Training Loss")
plt.xlabel("Epoch")
plt.ylabel("Training Loss")
plt.plot(vanilla_epochs, vanilla_loss, label="Baseline PT")
plt.plot(optimized_epochs, optimized_loss, label="SM-Training-Compiler-enhanced PT")
plt.legend()
plt.show()
```



Convergence Example Plot



Note: For this example, due to the larger batch size that can be accommodated by the SageMaker Training Compiler, the initial decrease in training loss is greater when the SageMaker Training Compiler is enabled

Evaluation

In [31]:
```
# Create table of summary results including loss, accuracy, f1 scor
e, precision, recall, and steps per second
# for both native and optimized training jobs

table = pd.DataFrame(
    [native_summary["eval"][-1], optimized_summary["eval"][-1]],
    index=["Baseline PT", "SM-Training-Compiler-enhanced PT"],
)
table.drop(columns=["eval_runtime", "eval_samples_per_second", "epo
ch"])
```

Out[31]:

	eval_loss	eval_accuracy	eval_f1	eval_precision	eval_recall	eval_steps_per_seco
Baseline PT	0.376621	0.928199	0.929485	0.917749	0.941525	3.5
SM-Training-Compiler-enhanced PT	0.358390	0.922619	0.924309	0.909091	0.940044	2.2

Clean up

Stop all training jobs launched if the jobs are still running.

In [32]:
```
import boto3

sm = boto3.client("sagemaker")


def stop_training_job(name):
    status = sm.describe_training_job(TrainingJobName=name)["TrainingJobStatus"]
    if status == "InProgress":
        sm.stop_training_job(TrainingJobName=name)


stop_training_job(native_estimator.latest_training_job.name)
stop_training_job(sm_training_compiler_estimator.latest_training_job.name)
```

Also, to find instructions on cleaning up resources, see Clean Up (https://docs.aws.amazon.com/sagemaker/latest/dg/ex1-cleanup.html) in the *Amazon SageMaker Developer Guide*.