



AWS
re:Invent

AIM412-R1

Deep learning applications using PyTorch, featuring Freshworks

Kris Skrinak

PSA, Global Machine
Learning Segment Lead
Amazon Web Services

Michael Suo

Software Engineer
PyTorch

Tarkeshwar Thakur


VP Engineering
Freshworks

The broadest and most complete set of machine learning capabilities



The deep learning Amazon machine image (AMI)

- With Amazon Elastic Inference



Deep Learning AMI (Ubuntu 16.04)

AWS Deep Learning AMI are built and optimized for building, training, debugging, and serving deep learning models in EC2 with popular frameworks such as TensorFlow, MXNet, PyTorch, Chainer, Keras, and more. Deep learning frameworks are installed in Conda environments to provide a reliable and isolated environment for practitioners. The AWS Deep ...

[More info](#)
[View Additional Details in AWS Marketplace](#)

Product Details

By	Amazon Web Services
Customer Rating	★★★★★ (9)
Latest Version	24.3
Base Operating System	Linux/Unix, Ubuntu 16.04
Delivery Method	64-bit (x86) Amazon Machine Image (AMI)
License Agreement	End User License Agreement
On Marketplace Since	11/14/17
AWS Services Required	Amazon EC2, Amazon EBS

Highlights

- Used Ubuntu 16.04 as base

Pricing Details

Hourly Fees

Instance Type	Software	EC2	Total
t2.small	\$0.00	\$0.023	\$0.023/hr
t2.medium	\$0.00	\$0.046	\$0.046/hr
t2.large	\$0.00	\$0.093	\$0.093/hr
t2.xlarge	\$0.00	\$0.186	\$0.186/hr
t2.2xlarge	\$0.00	\$0.371	\$0.371/hr
t3.small	\$0.00	\$0.021	\$0.021/hr
t3.medium	\$0.00	\$0.042	\$0.042/hr
t3.large	\$0.00	\$0.083	\$0.083/hr
t3.xlarge	\$0.00	\$0.166	\$0.166/hr
t3.2xlarge	\$0.00	\$0.333	\$0.333/hr
m5a.large	\$0.00	\$0.086	\$0.086/hr
m5a.xlarge	\$0.00	\$0.172	\$0.172/hr
m5a.2xlarge	\$0.00	\$0.344	\$0.344/hr
m5a.4xlarge	\$0.00	\$0.688	\$0.688/hr
m5a.12xlarge	\$0.00	\$2.064	\$2.064/hr
m5a.24xlarge	\$0.00	\$4.128	\$4.128/hr
m5d.large	\$0.00	\$0.113	\$0.113/hr
m5d.xlarge	\$0.00	\$0.226	\$0.226/hr

[Cancel](#) [Continue](#)

Training with PyTorch estimators

```
import os
import subprocess

instance_type = 'local'

if subprocess.call('nvidia-smi') == 0:
    ## Set type to GPU if one is present
    instance_type = 'local_gpu'

print("Instance type = " + instance_type)
```

```
from sagemaker.estimator import Estimator

hyperparameters = {'epochs': 1}

estimator = Estimator(role=role,
                      train_instance_count=1,
                      train_instance_type=instance_type,
                      image_name='pytorch-extending-our-containers-cifar10-example:latest',
                      hyperparameters=hyperparameters)

estimator.fit('file:///tmp/pytorch-example/cifar-10-data')

predictor = estimator.deploy(1, instance_type)
```

Deploying PyTorch at scale with Amazon SageMaker

```
from sagemaker.estimator import Estimator

hyperparameters = {'epochs': 1}

instance_type = 'ml.m4.xlarge'

estimator = Estimator(role=role,
                      train_instance_count=1,
                      train_instance_type=instance_type,
                      image_name=ecr_image,
                      hyperparameters=hyperparameters)

estimator.fit(data_location)

predictor = estimator.deploy(1, instance_type)

# get some test images
dataiter = iter(testloader)
images, labels = dataiter.next()

# print images
imshow(torchvision.utils.make_grid(images))
print('GroundTruth: ', ' '.join('%4s' % classes[labels[j]] for j in range(4)))

predictor.accept = 'application/json'
predictor.content_type = 'application/json'

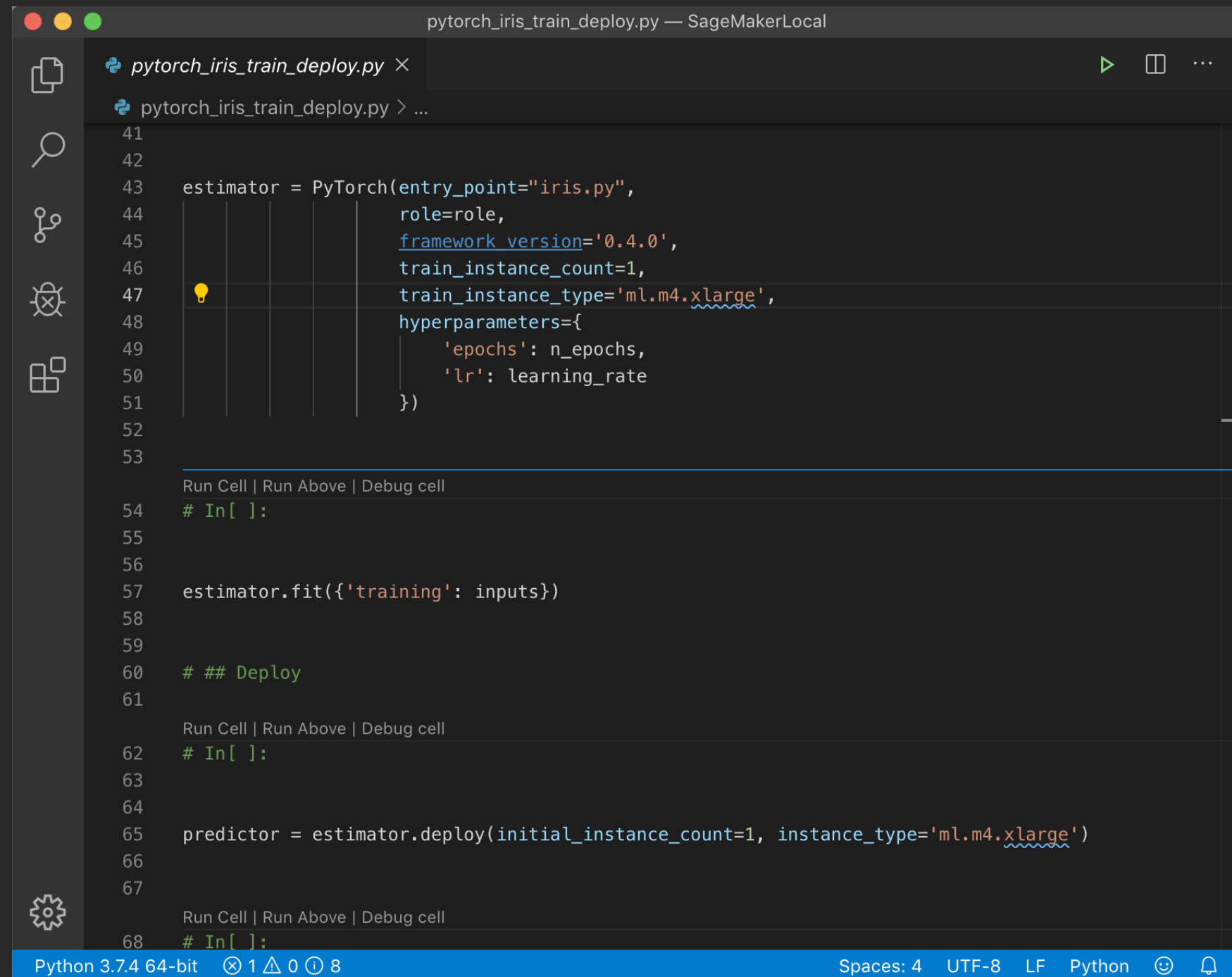
predictor.serializer = json_serializer
predictor.deserializer = json_deserializer

outputs = predictor.predict(images.numpy())

_, predicted = torch.max(torch.from_numpy(np.array(outputs)), 1)

print('Predicted: ', ' '.join('%4s' % classes[predicted[j]]
                              for j in range(4)))
```

PyTorch in Amazon SageMaker Local



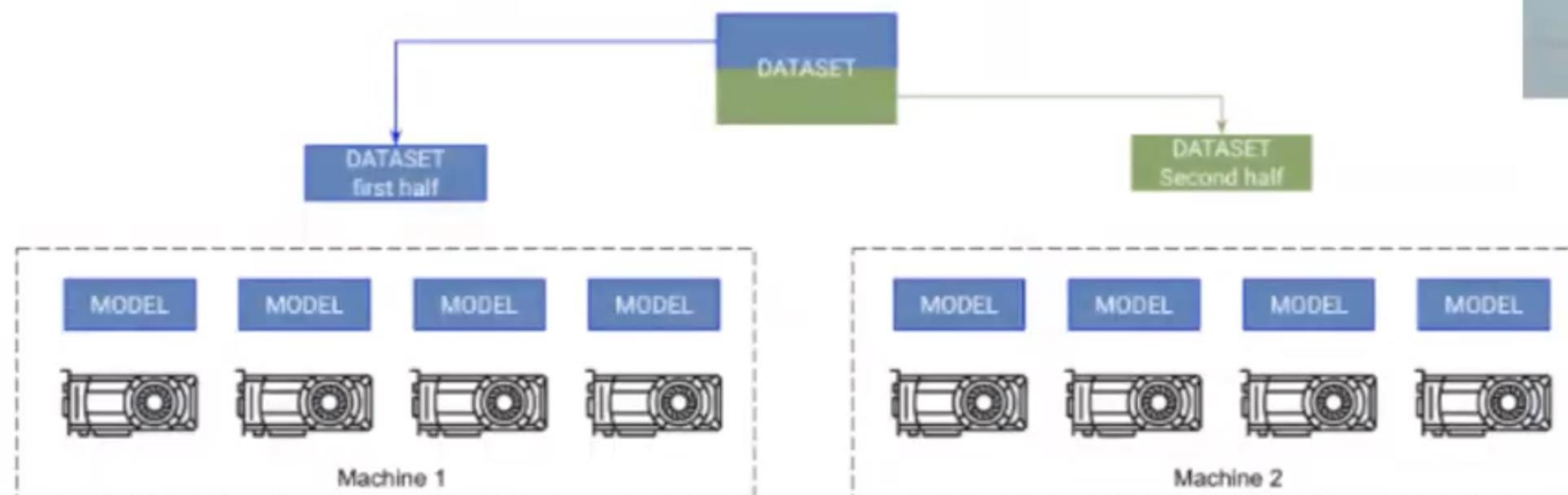
The screenshot shows a code editor window titled "pytorch_iris_train_deploy.py — SageMakerLocal". The editor contains a Python script for training and deploying a PyTorch model. The script is divided into three sections, each with a "Run Cell | Run Above | Debug cell" button. The first section (lines 41-53) defines a PyTorch estimator. The second section (lines 54-59) trains the estimator. The third section (lines 60-67) deploys the estimator. The script uses the SageMaker Python SDK.

```
41
42
43 estimator = PyTorch(entry_point="iris.py",
44                     role=role,
45                     framework_version='0.4.0',
46                     train_instance_count=1,
47                     train_instance_type='ml.m4.xlarge',
48                     hyperparameters={
49                         'epochs': n_epochs,
50                         'lr': learning_rate
51                     })
52
53
54 # In[ ]:
55
56
57 estimator.fit({'training': inputs})
58
59
60 # ## Deploy
61
62 # In[ ]:
63
64
65 predictor = estimator.deploy(initial_instance_count=1, instance_type='ml.m4.xlarge')
66
67
68 # In[ ]:
```

Python 3.7.4 64-bit 1 0 8 Spaces: 4 UTF-8 LF Python

PyTorch Lightning on AWS

Multi-node Training



Every GPU on every machine gets a copy of the model. Each machine gets a portion of the data and trains only on that portion. Each machine syncs gradients with the other.



Latest on PyTorch

Michael Suo
Software Engineer
PyTorch

What is PyTorch?



An open source deep learning platform

GPU-enabled Tensors with behaviour similar to NumPy

What is PyTorch?

Graphs are defined dynamically, as they are executed in Python

Fast tape-based autograd

A graph is created on the fly



```
W_h = torch.randn(20, 20, requires_grad=True)
W_x = torch.randn(20, 10, requires_grad=True)
x = torch.randn(1, 10)
prev_h = torch.randn(1, 20)
```



What is PyTorch?





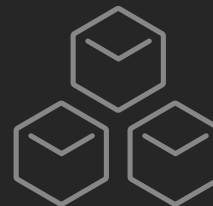
PYTORCH 1.3



NEW CORE FRAMEWORK FEATURES



NEW LIBRARIES



NEW FRAMEWORKS



PYTORCH 1.3

PYTORCH MOBILE

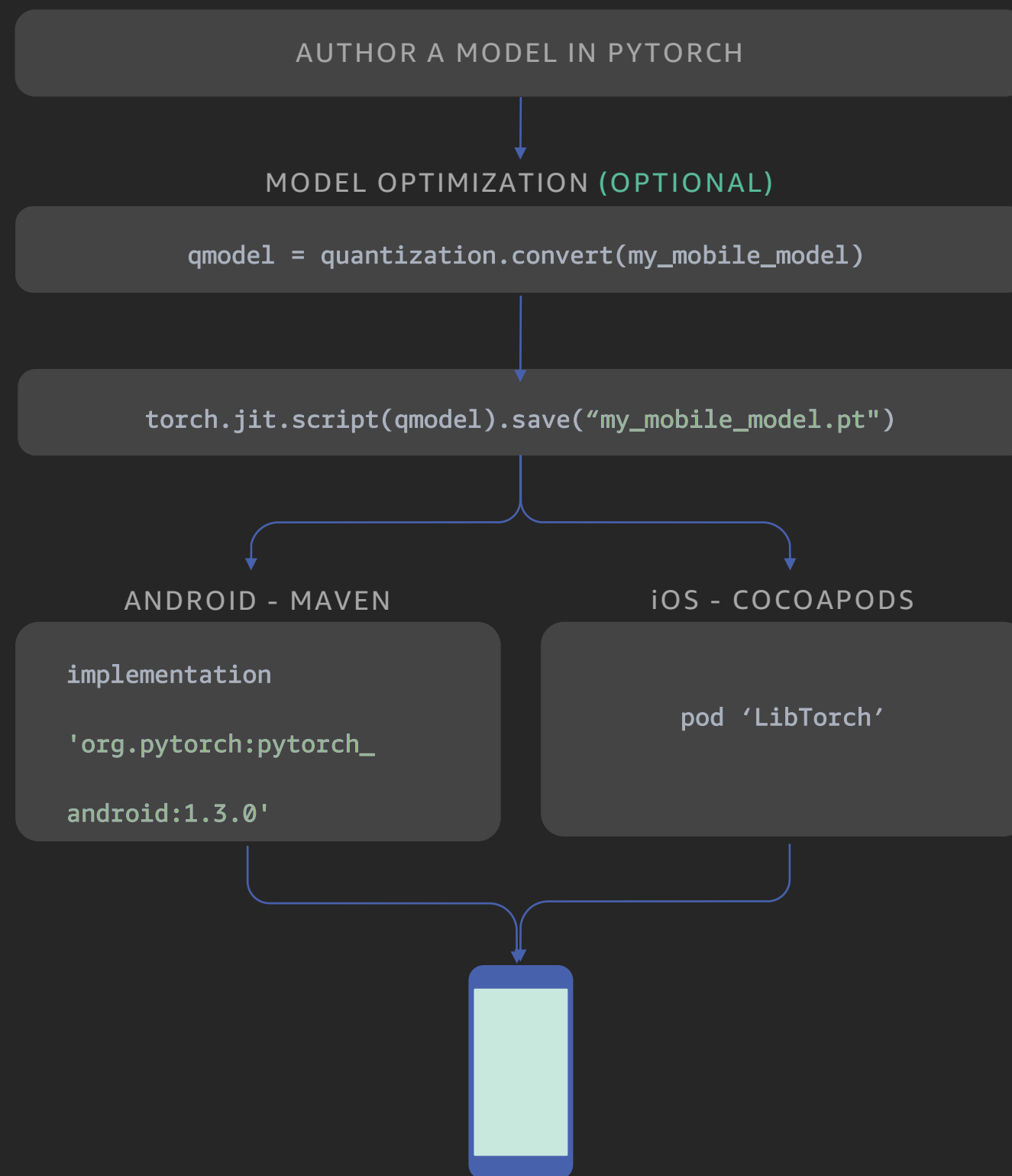
EXPERIMENTAL

End-to-end workflows for mobile in iOS
and Android:

- No separate runtime to export

COMING SOON

- Build-level optimization and selective compilation
- Whole program optimization with link time optimization





PYTORCH 1.3

QUANTIZATION

EXPERIMENTAL

- Neural networks inference is expensive
- IoT and mobile devices have limited resources
- Quantizing models enables efficient inference at scale

4X

LESS MEMORY
USAGE

2-4X

SPEEDUPS IN
COMPUTE

```
model = ResNet50()

model.load_state_dict(torch.load("model.pt"))

qmodel = quantization.prepare(
    model, {"": quantization.default_qconfig})

qmodel.eval()

for batch, target in data_loader:
    model(batch)

qmodel = quantization.convert(qmodel)
```



PYTORCH 1.3

NAMED TENSORS

EXPERIMENTAL

- Enables cleaner, better code with more expressivity — tensors are manipulated based on names
- Code becomes self-documenting by adding names to tensor dimensions
- Prevent silent user errors through runtime checking of names

```
# Tensor[N, C, H, W]
images = torch.randn(32, 3, 56, 56)
images.sum(dim=1)
images.select(dim=1, index=0)
```

Today we name and access dimensions by comment

```
NCHW = ['N', 'C', 'H', 'W']
images = torch.randn(32, 3, 56, 56, names=NCHW)
images.sum('C')
images.select('C', 0)
```

But naming explicitly leads to more readable and maintainable code



TORCHSCRIPT

A static, high-performance subset of Python.

1. Prototype your model with PyTorch
2. Control flow is preserved
3. First-class support for lists, dictionaries, etc.

```
import torch
class MyModule(torch.nn.Module):
    def __init__(self, N, M, state: List[Tensor]):
        super(MyModule, self).__init__()
        self.weight = torch.nn.Parameter(torch.rand(N, M))
        self.state = state

    def forward(self, input):
        self.state.append(input)
        if input.sum() > 0:
            output = self.weight.mv(input)
        else:
            output = self.weight + input
        return output

# Compile the model code to a static representation
my_module = MyModule(3, 4, [torch.rand(3, 4)])
my_script_module = torch.jit.script(my_module)

# Save the compiled code and model data
# so it can be loaded elsewhere
my_script_module.save("my_script_module.pt")
```



PYTORCH JIT

An optimizing just-in-time compiler
for PyTorch programs.

1. Lightweight, thread-safe interpreter
2. Easy to write custom transformations
3. Not just for inference! Autodiff support

```
graph(%self : ClassType<MyModule>,
      %input.1 : Tensor):
  %16 : int = prim::Constant[value=1]()
  %6 : None = prim::Constant()
  %8 : int = prim::Constant[value=0]()
  %2 : Tensor[] = prim::GetAttr[name="state"](%self)
  %4 : Tensor[] = aten::append(%2, %input.1)
  %7 : Tensor = aten::sum(%input.1, %6)
  %9 : Tensor = aten::gt(%7, %8)
  %10 : bool = aten::Bool(%9)
  %output : Tensor = prim::If(%10)
    block0():
      %11 : Tensor = prim::GetAttr[name="weight"](%self)
      %output.1 : Tensor = aten::mv(%11, %input.1)
      -> (%output.1)
    block1():
      %14 : Tensor = prim::GetAttr[name="weight"](%self)
      %output.2 : Tensor = aten::add(%14, %input.1, %16)
      -> (%output.2)
  return (%output)
```



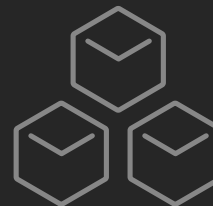

PYTORCH 1.3



NEW CORE FRAMEWORK FEATURES



NEW LIBRARIES



NEW FRAMEWORKS



PYTORCH 1.3

CRYPTEN

A platform for research in machine learning using secure-computation techniques

KEY FEATURES:

- Tensors and CryPTensors coexist and can be mixed and matched
- Uses standard eager execution — no compilers! Easy debugging and learning
- Support for secure multi-party computation (MPC)
- Homomorphic encryption (FHE) (COMING)
- Trusted execution environments (COMING)

```
import crypten
import torch

crypten.init() # sets up communication
x = torch.tensor([1.0, 2.0, 3.0])
x_enc = crypten.cryptensor(x) # encrypts tensor
x_dec = x_enc.get_plain_text() # decrypts tensor
assert torch.all_close(x_dec, x) # this passes!

y_enc = crypten.cryptensor([2.0, 3.0, 4.0])
xy_enc = x_enc + y_enc # adds encrypted tensors
xy_dec = xy_enc.get_plain_text()
assert torch.all_close(xy_dec, x + y) # this passes!

z = torch.tensor([4.0, 5.0, 6.0])
xz_enc = x_enc + z # adds FloatTensor to CrypTensor
xz_dec = xz_enc.get_plain_text()
assert torch.all_close(xz_dec, x + z) # this passes!
```



PYTORCH 1.3

CAPTUM

Model interpretability library for PyTorch

SUPPORT FOR ATTRIBUTION ALGORITHMS TO INTERPRET:

- Output predictions with respect to inputs
- Output predictions with respect to layers
- Neurons with respect to inputs
- Currently provides gradient-based approaches (e.g., Integrated Gradients)

Target Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
zebra	zebra (0.60)	zebra	7.54	what is on the picture



Text contributions: 7.54

Image contributions: 11.19

Total contributions: 18.73



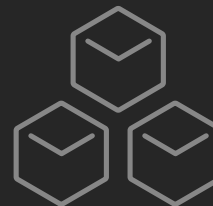
PYTORCH 1.3



NEW CORE FRAMEWORK FEATURES



NEW LIBRARIES



NEW FRAMEWORKS



PYTORCH 1.3

DETECTRON2

- Support for the latest models and new tasks
- Increased flexibility to aid computer vision research
- Improvements in maintainability and scalability to support production use cases





PYTORCH 1.3

SPEECH EXTENSIONS TO FAIRSEQ

Now supports end-to-end learning for speech recognition

NEW FEATURES INCLUDE:

- Attention-based and CTC-based approaches
- Transformer-based ASR models
- Integration of the Wav2Letter++ decoder into Fairseq
- Compatible with torchaudio and Kaldi for feature extraction

Branch: master ▾ fairseq / examples / speech_recognition / Create new file Upload files Find file History

okhonko and facebook-github-bot Fix method has same name as property ... Latest commit 4812f64 on Aug 20

..

criteria	Asr initial push (#810)	2 months ago
data	Fix method has same name as property	last month
datasets	Asr initial push (#810)	2 months ago
models	Small fixes	2 months ago
tasks	Asr initial push (#810)	2 months ago
README.md	Asr initial push (#810)	2 months ago
__init__.py	Asr initial push (#810)	2 months ago
infer.py	Asr initial push (#810)	2 months ago

README.md

Speech Recognition

`examples/speech_recognition` is implementing ASR task in Fairseq, along with needed features, datasets, models and loss functions to train and infer model described in [Transformers with convolutional context for ASR \(Abdelrahman Mohamed et al., 2019\)](#).

Additional dependencies

On top of main fairseq dependencies there are couple more additional requirements.

1. Please follow the instructions to install [torchaudio](#). This is required to compute audio fbank features.
2. [Sclite](#) is used to measure WER. Sclite can be downloaded and installed from source from sctk package [here](#). Training and inference doesn't require Sclite dependency.

Preparing librispeech data

```
./examples/speech_recognition/datasets/prepare-librispeech.sh $DIR_TO_SAVE_RAW_DATA $DIR_FOR_PREPROCESSED_
```

Training librispeech data

```
python train.py $DIR_FOR_PREPROCESSED_DATA --save-dir $MODEL_PATH --max-epoch 80 --task speech_recognition
```

Inference for librispeech

Resources



[PyTorch.org](https://pytorch.org)



Youtube.com/pytorch



Twitter.com/pytorch



Facebook.com/pytorch



Medium.com/pytorch

Freshworks

Tarkeshwar Thakur

VP Engineering
Freshworks

We build business software

that enables our customers to deliver
moments of wow

Our scale

150,000+

Global customers

3 Million+

Support tickets per day

375,000

Active support agents



To consider

Diversity

Customers in insurance, finance, travel, logistics, etc.

Uniqueness

Each customer has unique vocabulary, jargon, technical terms

Security

Need to keep customer information private and secure, no sharing

Meet Jonathan

System admin

Problem statement

Routing

Who is the best person to solve the customer's problem?

Categorization

What kind of issue is it?

Prioritization

How urgent is the issue?



The Goal

Reduce ticket assignment and resolution times
Improve triage and set context for agents

Ticket fields

Tickets assigned to me > 227189

☆ Reply Add Note Forward Merge Delete

✉ Please help! Issues with SAML setting in my account. Help needed urgently. #227189 META

Andrew Paul reported via email Created by Sudharshan Karthik

Overdue Customer Responded

Andrew Paul reported via email, 12 hrs ago (Wed, 26 Sep 2018 at 10:12 am) to: support@freshdesk.com; cc: sales@freshdesk.com

Hey there,

I have couple of issues. I trying to get a new instance with SAML SSO, https://freshdesk.com However, the ADMIN gui accepted the change but as you as you go back to the same screen, SSO is turned off and all the SSO is gone. Secondly, I got couple email indicating Sha256 change and I want to verify that we are already okay. Okta indicated that they already using sha256 for Freshdesk and want to confirm with you regarding existing instances below.

Here's my account login URL: https://support.googlework.co.in

Best,
Andrew Paul
Sr. Systems Engineer

Open

- FIRST RESPONSE DUE
by Mon 3 Jul 2018, 09:18pm
- RESOLUTION DUE [Edit](#)
by Mon 3 Jul 2018, 09:18pm

PROPERTIES ⚡ 3

Tags

L2 Bug × Forum App ×
Helpdesk Issues × +4

Status

Open

Priority

● Urgent

Group assigned to

--

Agent assigned to

--

Update

⚡ FREDDY PREDICTED FIELDS

- ✓ Priority : ● Urgent
- ✓ Group : L2 Support
- ✓ Category : Troubleshooting

CANCEL APPLY

What we needed

Build text classification models

Use ticket text and subject to predict fields

Quick, accurate, and small models

Chose Facebook's fastText & PyTorch: high accuracy, quick modeling times, quantization, extensibility

Faster model build times

Initial pipeline took 24–30 hours to build 35K models on a Spark cluster

Amazon SageMaker + PyTorch

Time taken to build 30k models significantly
reduced from 24 hours to 60 minutes

Why PyTorch

Hassle-free transition

Transition from custom pipeline → PyTorch containers on Amazon SageMaker was seamless

Extensibility to deeper networks

Wanted to experiment with modern deep learning architectures for embeddings & classifiers in PyTorch

Quantization, interpretability, and combining classifiers

Focus on smaller models, explaining predictions, and building multi-head/multi-label classifiers

Data sources

Data processing

Training

Inference

Consumption



Amazon S3



Amazon SageMaker



Endpoints interface API



Predictive model f_{π}



Endpoints serving models



Scalable cluster



Freshworks
suite



Data lake



Data processing, extraction, etc.



Predictive modeling f_{π}



Scalable cluster



Hadoop HDFS

APIs

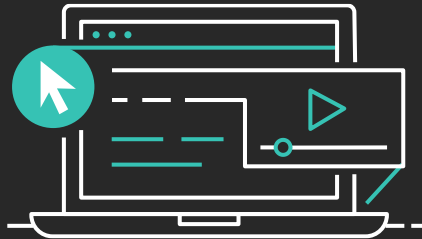


Learn ML with AWS Training and Certification

The same training that our own developers use, now available on demand



Role-based ML learning paths for developers, data scientists, data platform engineers, and business decision makers



70+ free digital ML courses from AWS experts let you learn from real-world challenges tackled at AWS



Validate expertise with the
AWS Certified Machine Learning - Specialty exam

Visit <https://aws.training/machinelearning>

Thank you!

Tarkeshwar Thakur

tarkeshwar.thakur@freshworks.com



Please complete the session
survey in the mobile app.