22nd Apr, 2022 175th Parallel Programming Workshop with Trial Account "Supercomputing for Beginners"

Running machine learning on supercomputers – for beginners

19th Apr 2022 v1.2







Machine learning



linear relation, multiple regression \rightarrow machine learning

an example in classification tasks



nonlinear relationship between the feature values and classes \rightarrow deep learning

4

Deep learning



input layer

backprop

propagate error by differentiation update weight in each layer

loss function

error estimate

output layer || target

$$y = \sum_{j} w_{j} x_{j}$$
$$w_{j} : weight$$

$$z = F(y)$$

activation function = nonlinear ReLU, Tanh, Sigmoid,...



Tools for machine learning

- majority = using Python (+ R, Octave, etc)
 - Why?
 - simple syntax and language \bullet
 - fast & convenient libraries for matrix operations, incl. numpy \bullet
 - many people use it :) software stacks, ecosystems lacksquare \checkmark we don't know how it will change after 10 years, of course...
- specialized frameworks for machine learning are recommended
 - neural network can be easily designed
 - fast autograd for backprops, optimization tools
 - runs fast on GPGPUs
 - easy parallel learning (important for ML + HPC) ✓ **PyTorch**, TensorFlow, Keras, Caffe, (Chainer)



machine learning and GPUs, supercomputers

- In ML/DL, most arithmetics are multiple-add matrix operations
 - affinity with 3D graphics

 - tens of times faster than CPUs.
- amount of GPUs are needed example) Google JFT-300M = 3 billion images
 - -

- GPUs are good at it, ML is one of the main targets of new GPU development.

• Machine learning is becoming bigger and bigger, sometimes a massive

NN models for huge-scale learning, such as Transformer, MLP-Mixer

supercomputers in private sectors — Selene@NVIDIA, MN3@PFN, and etc.



(reference) parallelization in deep learning

2 classes of methods for learning on multiple processors / GPUs.

Data parallel

Data divided, the network is copied over GPUs

- 1. update NN model with different data
- 2. take averages over weights



immediately available in many frameworks

Model parallel

divide model into different ranks automation tools being developed



expertise required



Exercise: python3 environment settings (1/5)

First, prepare your environment in /work directory.

[tUVXYZ@obcx01 ~]\$ cd /work/gt00/tUVXYZ/ [tUVXYZ@obcx01 tUVXYZ]\$ mkdir deeplearning [tUVXYZ@obcx01 tUVXYZ]\$ cd deeplearning Enter

It is important to know the current environment. Let us confirm the operating system (on OBCX) and the version of python.

[tUVXYZ@obcx01 deeplearning]\$ cat /proc/version Enter Linux version 3.10.0-957.21.3.el7.x86_64 (mockbuild@x86-017.build.eng.bos.redhat.com) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-36) (GCC)) #1 SMP Fri Jun 14 02:54:29 EDT 2019 [tUVXYZ@obcx01 deeplearning]\$ python --version Enter **Python 2.7.5** [tUVXYZ@obcx01 deeplearning]\$ python3 Enter -bash: python3: command not found

As we are using is Red Hat Enterprise Linux 7, the default python is v2.







Exercise: python3 environment settings (2/5)

 \star Many of the latest frameworks incl. PyTorch require Python 3. Therefore, we cannot use default python on OBCX.

<u>Cautions for environment settings on supercomputers</u>

• Users do not have root (administrator) privileges, unlike on personal computers.

- you need to work as non-root.

- The environment should be set up on the nodes that are connected to the internet. \bullet - on OBCX, therefore, we need to use the login node.
- We will run machine learning jobs on the compute nodes. libraries should be built (set up) on /work on OBCX.
- Updates in python libraries often lacks backward compatibility
 - we use pip, miniconda, and etc, in order to keep consistent versions.
- We may use virtual environments, if we do different machine learning with different tools.
 - you may practice today how to use pip virtualenv (Option B)
 - it is possible to use Docker containers even on supercomputers



Exercise: python3 environment settings (3/5)

First, load python3 environment on OBCX.

[tUVXYZ@obcx01 deeplearning]\$ module purge [tUVXYZ@obcx01 deeplearning]\$ module load python/3.7.3 [tUVXYZ@obcx01 deeplearning]\$ python3 --version Enter **Python 3.7.3**

Then we set up the package manager (pip3). You may choose one.

pip3 user install \leftarrow for a beginner **Option 1** Option 2 pip3 virtual env



- \leftarrow if you are familiar with python+ML



11

Exercise: python3 environment settings (4/5)

Option 1 - pip3 user install

set up the installation directory when --user option is used in pip3. set it "PYTHONUSERBASE" environment variable

[tUVXYZ@obcx01 deeplearning]\$ [tUVXYZ@obcx01 deeplearning]\$ echo \$PYTHONUSERBASE /work/gt00/t00570/deeplearning/.local

update pip3 (= package manager for python3) to the latest version.

[tUVXYZ@obcx01 deeplearning]\$ pip3 install --upgrade pip --user







Exercise: python3 environment settings (5/5)

Option 2 - pip3 virtual environment

create a virtual environment

[tUVXYZ@obcx01 deeplearning]\$ python3 -m venv torch_env

You will find a directory "torch_env", wherein the virtual environment is set up. We modify the script for activating the virtual environment as follows:

3.7.3/lib:\$LD_LIBRARY_PATH >> ./torch_env/bin/activate

Then we launch the virtual environment. You will have the prompt with its name. update pip3 (= package manager for python3) to the latest version.

[tUVXYZ@obcx01 deeplearning]\$ source ./torch_env/bin/activate (torch_env) [tUVXYZ@obcx01 deeplearning]\$ pip3 install --upgrade pip

- virtual environment = separated environment intended for a single specific application







Install PyTorch

Confirm which environments are needed and how to install on the website.

https://pytorch.org
>> "Get Started"
>> "Start Locally"

There are no GPUs on OBCX. We install PyTorch LTS (1.8.2) for CPUs.

O PyTorch

GET STARTED

Select preferences and run the command to install PyTorch locally, or get started quickly with one of the supported cloud platforms.

Start Locally

Shortcuts

Prerequisites

Python

Package Manager

Installation

Anaconda

pip

Verification

Building from source

Prerequisites

Start via Cloud Partners

Previous PyTorch Versions

Mobile

Supported Linux Distributions

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.12 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also install previous versions of PyTorch. Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the PyTorch Enterprise Support Program.





Installing PyTorch

Type the command as it is (copy from the PyTorch web site ,with no linebreaks!)

[tUVXYZ@obcx01 deeplearning]\$ pip3 install torch==1.8.2+cpu torchvision==0.9.2+cpu torchaudio==0.8.2 -f https://download.pytorch.org/whl/lts/1.8/torch_lts.html --user Enter

Confirm installed packages by "pip3 list"

Installation successful if you find "torch 1.8.2+cpu" in the list ("--user" is unnecessary for Option 2)

[tUVXYZ@o Package	bcx01 deeplear Version	rning]\$	pip3 list	Enter
numpy	1.21.6			
Pillow	9.1.0			
pip	22.0.4			
setuptools	40.8.0			
torch	1.8.2+cpu			
torchaudio	0.8.2			
torchvision	0.9.2+cpu			
typing_exte	nsions 4.1.1			



exercising machine learning with sample program (1/4)

Today, we will run image classification task of FashionMNIST on PyTorch tutorial https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html

<u>1-1 preparation of the dataset</u>

Compute nodes on OBCX are not connected to the internet \rightarrow We prepare the dataset in advance

[tUVXYZ@obcx01 deeplearning]\$ cp /work/gt00/share/z30122/torch_sample/download.py . Enter [tUVXYZ@obcx01 deeplearning]\$ python3 download.py Enter

download.py (excerpt)

training_data = torchvision.datasets.FashionMNIST(root="data", train=True, download=True, transform=torchvision.transforms.ToTensor()

The dataset is downloaded into the directory "data" (< 30 MB)



exercising machine learning with sample program (2/4)

<u>1-2. preparation of the python script</u>

We prepared a single Python script (in ful) as "samp.py". Copy it to your directory.

[tUVXYZ@obcx01 deeplearning]\$ cp /work/gt00/share/z30122/torch_sample/samp.py .

Then step forward to making your job scripts. (next page)





exercising machine learning with sample program (3/4)

2. write your job script

[tUVXYZ@obcx01 deeplearning]\$ emacs job.sh Enter

Option 1 (pip3 user install)

job.sh

```
#!/bin/bash
#PJM -L rscgrp=tutorial
#PJM -L node=1
#PJM -L elapse=0:15:00
#PJM -g gt00
#PJM -N fashionMNIST
#PJM -o result.txt
#PJM -j
module load python/3.7.3
export PYTHONUSERBASE=/work/gt00/tUVXYZ
/deeplearning/.local (no line break)
python3 samp.py
```

Option 2 (pip3 virtualenv)

job.sh

```
#!/bin/bash
#PJM -L rscgrp=tutorial
#PJM -L node=1
#PJM -L elapse=0:15:00
#PJM -g gt00
#PJM -N fashionMNIST
#PJM -o result.txt
#PJM -j
source ./torch_env/bin/activate
python3 samp.py
```



exercising machine learning with sample program (4/4)3. execute machine learning and confirm the result

submit the job

[tUVXYZ@obcx01 deeplearning]\$ pjsub job.sh

confirm the result

It is successful if you get the following result.

```
Using cpu device
Shape of X [N, C, H, W]: torch.Size([64, 1, 28, 28])
Shape of y: torch.Size([64]) torch.int64
NeuralNetwork(
  (flatten): Flatten(start_dim=1, end_dim=-1)
  (linear_relu_stack): Sequential(
    (0): Linear(in_features=784, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=10, bias=True)
```

[tUVXYZ@obcx01 deeplearning]\$ less result.txt

Epoch 1	Epoch 5								
loss: 2.300122 [0/60000] loss: 2.292114 [6400/60000]	loss: 1.343366 [0/60000] loss: 1.312548 [6400/60000]								
loss: 2.272950 [12800/60000]	loss: 1.152993 [12800/60000]								
loss: 2.271642 [19200/60000]	loss: 1.247405 [19200/60000]								
loss: 2.254692 [25600/60000]	loss: 1.125050 [25600/60000]								
loss: 2.223023 [32000/60000]	loss: 1.158031 [32000/60000]								
loss: 2.236395 [38400/60000]	loss: 1.172069 [38400/60000]								
loss: 2.201323 [44800/60000]	loss: 1.114118 [44800/60000]								
loss: 2.201231 [51200/60000]	loss: 1.147697 [51200/60000]								
loss: 2.168899 [57600/60000]	loss: 1.061013 [57600/60000]								
Test Error: Accuracy: 42.9%, Avg loss: 2.164772 	Test Error: Accuracy: 65.2%, Avg loss: 1.08492 Done!								

Enter

Enter



If you want to run your Jupyter Notebook on OBCX

[tUVXYZ@obcx01 deeplearning]\$ jupyter nbconvert --to python [FileName].ipynb

The above command generates [FileName].py file. Run it in your job script.

 On Wisteria/BDEC-01, JupyterHub interface is available, where Python script jobs can directly be submitted (to compute nodes) from your Jupyter Notebook.

	••	I -	$\langle \rangle$				ŵ				localhost.9888,	/lab/tree/Fa	shionMNIS	Lipynb	a g &	0	낢	0 Ó	+ 8	8			
										0	Coogle Calendar	Coogle map	ALC D	ropibes v									
					0	Start Locally	Py Iorch								8) ⊨ashio	nVNSL.	- Jugyteri ai	5				
С	File	Edit	View Run	Kernel	Tabs	Settings	Help																
-	ĽL	auncher		×	💌 Fas	hionMNIS	T.ipynb		•														
_	a	+ *	"D D	н Ш	GM	Code	~													e Pytho	n 3 (ipykerne	el) 🔿	
0		(a) a	demonst t	o nah																		- 1	1
-		[*]:	from tor	ch inpo	rt nn																		
≔			from tor	ch.util	s.data n impo	import rt datas	DataLo ets	ader															
			from tor	chvisic	n.tran	sforms i	mport	ToTen	sor														
*			# Downlo	ad trai	ning d	ata from	open	datas	ets.														
			training root	_data = ="data"	datası ,	ets.Fash	100MNI	STL															
			trai down	in=True, load=Tr	ue.																		
			tran	isform=1	oTenso	r(),																	
			<i>'</i>																				
			# Downlo test dat	ad test: a = dat	data : asets.	from ope FashionM	n dəta NIST(sets.															
			root	="data"	,																		
			down	load=Tr	ue,																		
) tran	isform=1	oTenso	r(),																	
			-																				
		[*]:	def trai	in(datal	oader, datalm	model, ader.dat	loss_fi aset)	n, opt	timizer):														
			mode	l trair	()																		
			for	batch, X, y =	(X, y) X. <mark>to</mark> (d	in chum evice),	erate() y. <mark>to</mark> (d	datalo evice)	oader):)														
				# Conne	te pre	diction	ecror																
				pred =	model()	X)																	
				LOSS =	LOSS_T	n(pred,	y)																
				# Backg optiniz	ropaga: er.zen	tion g_grad()																	
				loss.ba	ckward	0																	
				opriniz	eriste	111																	
				if bate los	:h % 100 s, cur	0 == 0: rent = l	oss.it	em().	batch * ler	n(X)													
				pri	nt(f"l	oss: {lo	ss:>7f	} [{	current:>5d	}/{si	ze:>5d}]")												
		[*]:	def test	(datalo	ader, i	rodel, l	oss_fn):															
			size	e = len(hatcher	datalo	ader.dat	aset)																
			mode	l eval)	Jacatoa	uer)																
	Simple		test	iloss.	connect three 2. G	t = 0.0	idle			-							Mode	Common	1 121	In 1 Col 1	Enchinet	NIST	urat
	annae		V 9.	\$ P	តាណាងព្រ	PAren en l	AGINE										WOOB:	Gormani		in tear	ratariturity	ana rup	ALUC

