



## Chapter 5.2: Running PyTorch On ROCm

ROCm Tutorial | AMD 2020

---



## Table of Contents

<b>CHAPTER 5.2: RUNNING PYTORCH ON ROCM</b> .....	<b>2</b>
PREPARATION.....	2
RUNNING THE EXAMPLE.....	2

## Chapter 5.2: Running PyTorch On ROCm

This hands-on tutorial shows how we can get started with the popular PyTorch framework as well as train a LSTM neural network on the Wikiext-2 dataset on a ROCm platform.

### Preparation

1. We will be using the AMD provided PyTorch docker container for this tutorial
  - Please ensure that ROCm is correctly installed and docker is also set up using the instructions in our installation guide
2. Obtain the PyTorch docker image corresponding to your ROCm installation:
  - We are using ROCm3.5
  - `docker pull rocm/pytorch:rocm3.5_ubuntu16.04_py3.6_pytorch`
3. Start the docker container:
  - `sudo docker run -it -v $HOME:/data --privileged --rm --device=/dev/kfd --device=/dev/dri --group-add video rocm/pytorch:rocm3.5_ubuntu16.04_py3.6_pytorch`
4. Install the torchvision library:
  - `pip3 install torchvision`

### Running the Example

Similar to the previous two examples, run the scripts “hipexamine-perl.sh” and “hipconvertinplace-perl.sh” to scan and port the files from CUDA to HIP using the inbuilt conversion tools

1. Clone the pytorch example inside your docker container:
  - `git clone https://github.com/pytorch/examples`
  - `cd examples/word_language_model`
2. The model for training the language model using the LSTM network is in `main.py`. Run the model:
  - `python3.6 main.py --cuda`
  - The script will start training the model

epoch	1		400/ 2983	batches		lr 20.00		ms/batch 59.16		loss 6.85		ppl 946.38
epoch	1		600/ 2983	batches		lr 20.00		ms/batch 59.04		loss 6.48		ppl 652.42
epoch	1		800/ 2983	batches		lr 20.00		ms/batch 59.14		loss 6.29		ppl 541.70
epoch	1		1000/ 2983	batches		lr 20.00		ms/batch 59.18		loss 6.15		ppl 467.59
epoch	1		1200/ 2983	batches		lr 20.00		ms/batch 59.44		loss 6.07		ppl 431.56
epoch	1		1400/ 2983	batches		lr 20.00		ms/batch 59.72		loss 5.94		ppl 381.58
epoch	1		1600/ 2983	batches		lr 20.00		ms/batch 59.85		loss 5.95		ppl 383.06
epoch	1		1800/ 2983	batches		lr 20.00		ms/batch 60.03		loss 5.80		ppl 330.85
epoch	1		2000/ 2983	batches		lr 20.00		ms/batch 60.16		loss 5.78		ppl 323.07
epoch	1		2200/ 2983	batches		lr 20.00		ms/batch 60.32		loss 5.66		ppl 288.09
epoch	1		2400/ 2983	batches		lr 20.00		ms/batch 60.96		loss 5.67		ppl 290.49
epoch	1		2600/ 2983	batches		lr 20.00		ms/batch 61.33		loss 5.65		ppl 285.71
epoch	1		2800/ 2983	batches		lr 20.00		ms/batch 61.37		loss 5.54		ppl 253.52
-----												
end of epoch	1		time: 189.28s		valid loss 5.55		valid ppl 257.45					
-----												
□												

**Figure 1: Output of the PyTorch LSTM execution on the Wikiext-2 dataset**