

ROCm pytorch

Used this tutorial to install pytorch for rocm, however I checked out release 1.5.

<https://github.com/ROCmSoftwarePlatform/pytorch/wiki/Building-PyTorch-for-ROCm> Allennlp was version 0.9.

GRU

BERT

This used bert-base with a batch size of 8.

Vega FE notes

The vega frontier edition results were obtained from a rented gpueater instance.

A batch size of 16 was also tried for the vega frontier edition to see if it would fit in vram and strangely the time per epoch dropped (01:12) with the larger batch size). This was also with thermal throttling as the vega fe was hitting 87 C and the clocks were down to 1.2 Ghz from 1.6 Ghz. The fans were limited to 40% under load on gpueater.com. It would be interesting to see what the performance is like with better thermals.

GPU	BERT-base emotion regression	GRU pos-tagger (1-hid)	GRU pos-tagger (2-hid)
GTX 1070	1:26.96	0:04.2	0:04.3
Tesla M40	1:32.76	0:04.05	0:04.3
RTX 3090	0:26.2	0:02.0	0:02.6
RX580	2:14.4	0:06.9	0:08.5
Vega Frontier	1:29.3	0:04.4	0:05.1
Vega Frontier (90% fans)	1:09.1	0:02.3	0:03.0

GPU	BERT-base emotion regression	GRU pos-tagger (1-hid)	GRU pos-tagger (2-hid)
Vega frontier (rocm 4.0)	1:07.5	0:02.4	0:02.9
i7-7800x	x	00:18	00:23
i9-7900x (defective?)	x	00:19	00:23
i9-7900x	x	00:16	00:20
i9-7980xe	x	00:15	00:18
e5-2680v3	x	00:27	00:34

using rocm apex gave no discernable performance improvement (with use_apex = true) However, it did reduce memory consumption by ~1GB for a batch of 16.

The RTX 3090 was tested with cuda 11, all other nvidia gpus were using cuda 10.2 (the RTX 3090 is not supported in this earlier version of cuda).

Revision #27

Created Fri, May 22, 2020 3:27 PM by [kenneth](#)

Updated Mon, Mar 22, 2021 6:56 PM by [kenneth](#)