

SpaCy GPU

Set Up Environment

It's relatively easy to use SpaCy with a GPU these days.

First set up your conda environment and install cudatoolkit (use nvidia-smi to match versions of the toolkit with the drivers):

Run `nvidia-smi`:

```
+-----+
| NVIDIA-SMI 440.100      Driver Version: 440.100      CUDA Version: 10.2  |
+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|   0   GeForce GTX 108...    Off   | 00000000:07:00.0 Off  |             N/A     |
| 28%   32C    P8      9W / 250W | 2427MiB / 11178MiB |    0%      Default  |
+-----+-----+-----+-----+-----+
|   1   GeForce GTX 108...    Off   | 00000000:09:00.0 Off  |             N/A     |
| 28%   28C    P8      8W / 250W | 29MiB / 11178MiB  |    0%      Default  |
+-----+-----+-----+-----+-----+
```

Create conda env:

```
conda create -n test python=3.8
conda activate test
conda install pytorch cudatoolkit=10.2 -c pytorch
```

Installing SpaCy

Now install spacy - depending on how you like to manage your python environments either carry on using conda for everything or switch to your preferred package manager at this point.

```
conda install -c conda-forge spacy cupy
```

or

```
pdm add 'spacy[cuda-autodetect]'
```

Download Models

Download a spacy transformer model to make use of your GPU/CUDA setup:

```
python -m spacy download en_core_web_trf
```

Using GPU

As soon as your code loads you should use the `prefer_gpu()` or `require_gpu()` functions to tell spacy to load cupy then load your model:

```
import spacy

spacy.require_gpu()

nlp = spacy.load('en_core_web_trf')
```

Now you can use the model to do some stuff

```
doc = nlp("My name is Wolfgang and I live in Berlin")

for ent in doc.ents:
    print(ent.text, ent.label_)
```

You can check that the GPU is actually in use with `nvidia-smi`:

```
Thu Oct 20 09:14:26 2022
+-----+
| NVIDIA-SMI 440.100      Driver Version: 440.100      CUDA Version: 10.2      |
+-----+-----+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf         Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+
|  0   GeForce GTX 108...   Off          | 00000000:07:00.0 Off  |          N/A         |
| 28%   31C    P8          8W / 250W  | 1149MiB / 11178MiB |    0%      Default   |
+-----+-----+-----+
|  1   GeForce GTX 108...   Off          | 00000000:09:00.0 Off  |          N/A         |
| 28%   27C    P8          8W / 250W  |  29MiB / 11178MiB |    0%      Default   |
+-----+-----+-----+

+-----+-----+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID    Type   Process name                                             Usage   |
+-----+-----+-----+
|    0         9770    C     ...scroft/miniconda3/envs/lbner/bin/python             1137MiB |
|    1         2752    G     /usr/lib/xorg/Xorg                                       9MiB   |
|    1         2852    G     /usr/bin/gnome-shell                                    6MiB   |
+-----+-----+-----+

(base) iravenscroft@shockwave:~/lbner test$ █
```

Also if you try to use transformer models without a GPU it will hang for AGES and max out your CPUs - another tell that something's not quite right.

Revision #2

Created 20 October 2022 08:19:13 by James

Updated 11 February 2024 16:27:02 by James