Week 5 Discussion

2 8

Dave DeBarr

All Sections

Search entries or author | Unread | Subscribe

Reply

Dave DeBarr

May 8, 2021

Bonus Question

[not for credit but relevant for work]

What's the maximum batch size you can use when finetuning the entire EfficientNetB0 network?

Why?

Edited by Dave DeBarr on May 8 at 7:39pm

Reply

Luis Garduno

May 9, 2021

I think it would be 100,000 since that's the number of training tensors?

Reply

Dave DeBarr

May 9, 2021

LOL: good point!

So what I meant to ask was, "What's the maximum batch size you can use when finetuning the entire EfficientNetB0 network on our K80 GPU?" [though responding for other GPUs is perfectably acceptable].

If you comment out the first ".fit()" statement in the homework and add "batchsize = 128" to the second ".fit()" statement, what happens?

Reply

Luis Garduno

May 9, 2021

Ah, got it. Yeah, using batch_size = 128 causes an OOM error.

With batch_size = 64, we should use around 19.75GB of memory:
((224 * 224 * 3 + 1) * 512 + 100001) x 4 x 64 = 19,755,737,344

So unless I'm missing something and taking into account that the K80 is supposed to have 24GB of memory, that would approximately give us a maximum batch_size of 25,769,803,776 x 64 / 19,755,737,344 = 83.48?

Edited by Luis Garduno on May 9 at 11:40pm

Reply

Dave DeBarr

May 10, 2021

Technically, the lab machines have a half of an Nvidia K80 GPU card: NC-series - Azure Virtual Machines | Microsoft Docs [search for "half"]; so you're getting 1 GPU with 12 gigabytes of memory.

The EfficientNetB0 model itself is not too large: 4,808,043 parameters (with adam optimizer state tripling this); but keeping the activations in memory dominates GPU memory usage.

If we replace "None" in the output shapes from the model summary with 1, and multiply to find the total number of values, we get something like 25,440,648 values. We multiply this value by 4 to convert values to bytes for float32 values, then multiply by the batch size to estimate the memory consumed.

So it looks like a batch size of 96 is okay (25440648 * 4 * 96 = 9,769,208,832; which is less than 12 gigabytes); but a batch size of 128 will fail (25440648 * 4 *

128 = 13,025,611,776; which is more than 12 gigabytes).

Edited by Dave DeBarr on May 10 at 9:40pm

Reply (1 like)

Hans-Erik Andersen

May 12, 2021

Hi Dave,

I had a follow-up question re. ConvNets for computer vision if you don't mind-

Most (all?) of the examples we've looked at involve image classification or segmentation with a single image (perhaps multi-channel). I'm wondering about a problem where you have different types of imagery for the same subject - does it make sense to stack the different types of images together as input to a ConvNet? And if the images are different resolutions, we would need to resample the imagery to a common resolution? For example, in a land cover classification application, we might have 2.5-meter (6-band) satellite imagery and 1-meter interferometric radar height measurements. Both of these sources of imagery are potentially useful for classifying land cover, but they are very different (i.e. passive vs. active remote sensing, etc.). Is it as simple as resampling all input image sources to a common resolution (e.g. 1-meter), stacking them together as different channels in a single e.g. 30mx30m input image (for example, 6 spectral bands + 1 IFSAR height) and feed into a ConvNet to classify land cover? Thanks!

Reply

Dave DeBarr

May 12, 2021

For the record: it's possible to create a multi-view convolutional neural network, where multiple input "images" are fed to a model. It might even be worth considering this as a possibility.

That said, it sounds like you have a satellite that is capturing data from multiple bands for the same region, just some bands have a different resolution (size for the area covered by a single pixel). Is this correct?

If so, then yeah; it's definitely possible to resize to a common size and feed the bands as channels to a convnet.

You may also want to consider alternatives for band/channel selection.

Interferometric Synthetic Aperture Radar (IfSAR): that's just fun to say :)

Reply

Hans-Erik Andersen

May 13, 2021

Thanks!! The multi-view convolutional neural network sounds really interesting -- for example, one could use a stack of images, all taken from different angles on a subject (or maybe a video taken from walking around a subject?) Or perhaps several overlapping aerial images, where each image is a slightly different perspective (stereo imagery).

Reply