	:
Week 4 Discussion  Dave DeBarr  All Sections	11 18
Search entries or author  Unread  In I	✓ <u>Subscribe</u>
← <u>Reply</u>	
Glen A Frison Apr 29, 2021  I think I had my terminal time out given long run time. Anyone else have this problem? I thought about it but figured the smake it not time out. But now regretting not asking before burning VM time. Using putty.	: spewing of output would
Glen A Frison Apr 29, 2021  I had already started the next run attempt, but trying this:  https://blog.webhostpython.com/2016/01/14/how-to-prevent-an-ssh-timeout-in-putty/	:
Hope it doesn't matter that I set it after logging in and starting.  CREDIX  Glen A Frison Apr 29, 2021  That worked.  Reply	:
Cool. There are a couple of other options as well:  1. Run the job as a background process, so it should keep running if you disconnect [no space between the '2' a nohup python cifar10-train.py.txt > stdout.txt 2> stderr.txt &  This redirects the standard output stream to stdout.txt and the standard error stream to stderr.txt.  2. Use something like the tmux (terminal multiplexer) so you can reconnect to the console later.  To install:  sudo apt install tmux	and the '>']:

tmux

Launch your command in the tmux console window; e.g. python cifar10-train.py.txt > stdout.txt 2> stderr.txt To detach from the tmux console: control+B [press and hold control key; press and release the 'b' key; then release the control key]; then press the To reattach to the tmux console for an existing session: tmux attach Please remember to turn off your machine after you're done using it. <u>Reply</u> 

√

√

√

(1 like) 0 Soo Lee Apr 30, 2021 Hi Dave, Could you let me know how to run the python code through Azure Notebook or Google Colab? ← Reply 💍 Dave DeBarr Apr 30, 2021 Hi Soo, I really don't spend much time with either of those options, though maybe someone else might chime in? For what it's worth: here are links to the sites ... https://notebooks\_azure\_com/ https://colab\_research\_google\_com/ It should be possible to run a notebook server from your Azure VM. Is this what you would like to do? ← Reply 스 Soo Lee 0 May 1, 2021 Thanks for your quick reply. !pip install kaggle from google.colab import files uploaded = files.upload() for fn in uploaded keys(): print('User uploaded file "{name}" with length {length} bytes'.format(

I tried to install the Kaggle API in Colab, but I had some problems as my Kaggle id is different from Google id. I solved this problem by downloading data to my laptop and uploading it again to Google Drive.

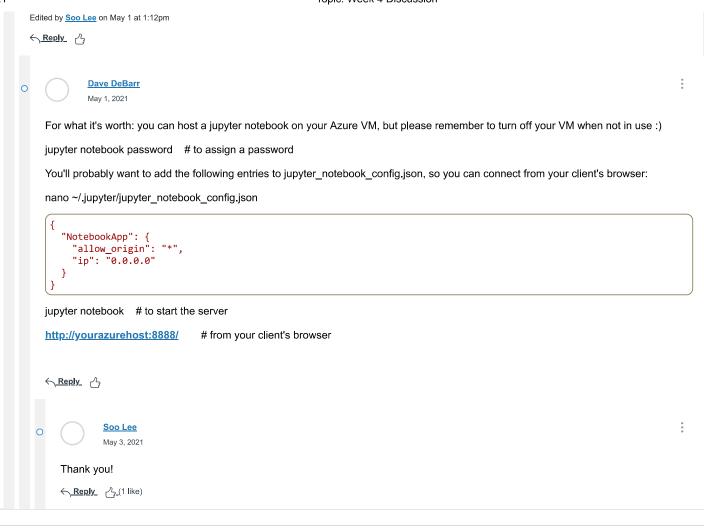
But I want to use the Azure Notebook and Kaggle API to download the data and study python code. I'm just wondering if I can use Azure Notebook like Google Colab.

!mkdir -p ~/.kaggle/ && mv kaggle.json ~/.kaggle/ && chmod 600 ~/.kaggle/kaggle.json

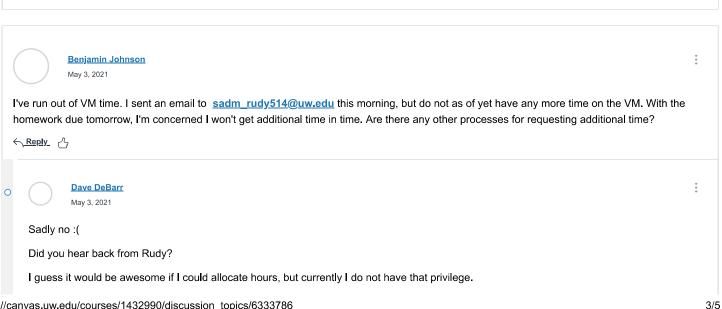
If anyone knows this, please let me know. Thanks:)

name=fn, length=len(uploaded[fn])))

!kaggle competitions download -c ml530-2021-sp-mnist







		I'll check with Rudy, but it could be about the same response time.		
		← <u>Reply</u> C		
	C	Benjamin Johnson May 4, 2021  OK, thanks. No, I've not heard from him yet. I actually finished the Friday, but unfortunately forgot to turn it off (mea maxima culpa). So the assignment is sitting there, complete, with no way for me to access it.		
	C	Benjamin Johnson May 4, 2021		
		Got additional time on my VM- thanks for your help.		
		<u>← Reply</u> <u>^3</u> (1 like)		
0		Dominic Maes May 4, 2021		
	۱h	have a general question that we may or may not have already addressed (maybe I missed it).		
	dit so	ets say we are building an image recognition model. When composing a training dataset of images, how do you handle the issue of having many ifferent images of different sizes? In order to build a model you will need have the same dimensions across all images correct? Would you apply ome type of up or down sampling of images to make them all the same size?		
		<u>Reply.</u>		
	0	Dave DeBarr May 4, 2021		
		On page 173 of the "Deep Learning with Python" textbook it says, "Consider the image of two African elephants shown in figure 5.34 (under a Creative Commons license), possibly a mother and her calf, strolling on the savanna. Let's convert this image into something the VGG16 model can read: the model was trained on images of size 224 × 224, preprocessed according to a few rules that are packaged in the utility function keras.applications.vgg16.preprocess_input. So you need to load the image, resize it to 224 × 224, convert it to a Numpy float32 tensor, and apply these preprocessing rules."		
		To make matters just a wee bit more complicated, there is an extra hyperparameter when resizing images: the "interpolation" method. For example, consider the interpolation methods found on this page: <a href="https://keras.jo/api/preprocessing/image/">https://keras.jo/api/preprocessing/image/</a> (bilinear, nearest, bicubic, area lanczos3, lanczos5, gaussian, mitchellcubic). The default is often sufficient; but if you're going to make a living with computer vision, you should consider exploring this hyperparameter too [notice how the default value for interpolation varies between the methods).		
		We'll also look at an example this evening where we endeavor to preserve the aspect ratio.		
		← <u>Reply</u> <sup>2</sup>		
0		Jacob van Baalen		
		May 6, 2021		
	I noticed in the homework for cifar10 that we are centering the data as well as scaling. Is there a reason that we centered in this case but not the others?			
	4	<u>_Reply</u>		
	0	Dave DeBarr May 6, 2021		
		Week 4? So last week :)		

Yup; nice catch!

"trnX\_mean = np.mean(trnX, axis = 0)" uses the training images to derive the mean pixel value for each row, column, and channel (red, green, or blue); so the shape of trnX\_mean is (32, 32, 3).

We then "center" the images by subtracting the mean values (after dividing the pixel values by 255), so the mean pixel value for each of the 32\*32\*3=3072 pixels for the training set is zero. This yields a mix of both positive and negative values in our input data.

For EfficientNet, they have actually wired normalization into the layers of their network:

https://github.com/tensorflow/blob/v2.4.1/tensorflow/python/keras/applications/efficientnet.py#L316-L317

How we preprocess our data is technically another hyperparameter for our image/data processing pipeline; e.g. see <a href="https://github.com/kerastuner/blob/master/kerastuner/applications/augment.py">https://github.com/kerastuner/applications/augment.py</a>.

Edited by <a href="Dave DeBarr">Dave DeBarr</a> on May 6 at 7:02pm

← Reply 💍