

# overcome Graphdef cannot be larger than 2GB in tensorflow

[Ask Question](#)


I am using tensorflow's [imageNet trained model](#) to extract the last pooling layer's features as representation vectors for a new dataset of images.

10

The model as is predicts on a new image as follows:

```
python classify_image.py --image_file new_image.jpeg
```

1

I edited the main function so that I can take a folder of images and return the prediction on all images at once and write the feature vectors in a csv file. Here is how I did that:

```
def main(_):
    maybe_download_and_extract()
    #image = (FLAGS.image_file if FLAGS.image_file else
    #         os.path.join(FLAGS.model_dir, 'cropped_panda.jpg'))
    #edit to take a directory of image files instead of a one file
    if FLAGS.data_folder:
        images_folder=FLAGS.data_folder
        list_of_images = os.listdir(images_folder)
    else:
        raise ValueError("Please specify image folder")

    with open("feature_data.csv", "wb") as f:
        feature_writer = csv.writer(f, delimiter='|')

        for image in list_of_images:
            print(image)
            current_features = run_inference_on_image(images_folder+"/"+image)
            feature_writer.writerow([image]+current_features)
```

It worked just fine for around 21 images but then crashed with the following error:

```
File "/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/ops.py", line 191
    raise ValueError("GraphDef cannot be larger than 2GB.")
ValueError: GraphDef cannot be larger than 2GB.
```

I thought by calling the method `run_inference_on_image(images_folder+"/"+image)` the previous image data would be overwritten to only consider the new image data, which doesn't seem to be the case. How to resolve this issue?

[python](#) [tensorflow](#)

share improve this question

edited Apr 12 at 16:00

| [user3666197](#)

asked Apr 1 '16 at 5:56

| [MedAli](#)



9,760 2 18 36



1,410 20 45

[add a comment](#)

### 3 Answers

[active](#)[oldest](#)[votes](#)

The problem here is that each call to `run_inference_on_image()` *adds* nodes to the same graph, which eventually exceeds the maximum size. There are at least two ways to fix this:

13

1. The **easy but slow** way is to use a different default graph for each call to `run_inference_on_image()` :

```
for image in list_of_images:
    # ...
    with tf.Graph().as_default():
        current_features = run_inference_on_image(images_folder+"/"+image)
    # ...
```

2. The **more involved but more efficient** way is to modify `run_inference_on_image()` to run on multiple images. Relocate your `for` loop to surround [this `sess.run\(\)` call](#), and you will no longer have to reconstruct the entire model on each call, which should make processing each image much faster.

[share](#) [improve this answer](#)

edited Aug 9 '16 at 18:44

answered Apr 1 '16 at 14:28



mrry

47.6k

3

115

165

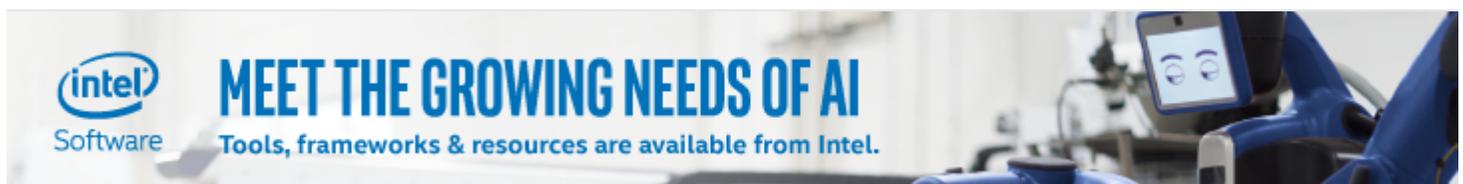
- 1 I went with the second option and it's faster. Thanks for idea! – [MedAli](#) Apr 1 '16 at 17:04

one question though, is there a way to pass an array of images instead of only one in the prediction part of the `sess.run` `predictions = sess.run(pool_3_tensor, {'DecodeJpeg/contents:0': image_data})` – [MedAli](#) Apr 1 '16 at 17:06

- 1 I think that particular feed point only works on a single image. It would be possible to change the graph so that it took a batch of images, but this would require creating a prefetching thread (using e.g. `tf.train.batch()`) to combine the images into a batch (which would have to all have the same size), and then feed into a slightly later point in the network. You would have to use the `input_map` argument to `tf.import_graph_def()` to change the tensor that is used as input. Since the structure of that particular graph is undocumented, it might be challenging though... – [mrry](#) Apr 1 '16 at 17:13

that's brilliant! thank you. – [MedAli](#) Apr 1 '16 at 18:26

Hello, can you explain what do you mean by surround the `sess.run` call? i've tried many different variations but i still run into the same error. thanks! :) – [Wboy](#) Jun 24 '16 at 14:07

[show 1 more comment](#)

You can move the `create_graph()` to somewhere *before* this loop `for image in list_of_images:` (which loops over files).

2 What it does is performing inference multiple times on the same graph.

share improve this answer

edited Feb 9 at 23:23

answered Dec 22 '16 at 20:51



[anh\\_ng8](#)

542 5 14

add a comment

---

The simplest way is put `create_graph()` at the first of main function. Then, it just create graph only

0 share improve this answer

edited Mar 27 at 17:24

answered Mar 27 at 17:01



[Div](#)

2,794 9 14 34



[SeokHyun Hwang](#)

1 1