**JetsonNX/Jetson Nano TensorRT Donkey Setup**

**1. Converting a .h5 file to a tensorrt directory**

a. Check that your Jetson or Docker Container has access to the GPUs

python

>>import tensorflow as tf

>>print("Available GPUs:", tf.config.list\_physical\_devices('GPU'))

If there is no GPU access, TensorRT can not be used. If the Docker can not access the GPU, you may need to make another container with the correct GPU flag in the bash script.

b. Once we know that there is GPU access we can continue to convert our .h5 model to a .savedmodel by running this code:

python

>>import tensorflow as tf

>>h5\_model\_path = “models/yourmodel.h5”

>>saved\_model\_path = “models/yourmodel\_converted.savedmodel”

>>model = tf.keras.models.load\_model(h5\_model\_path, compile=False)

>>model.save(saved\_model\_path)

>>exit()

Now we have a .savedmodel file.

**2. We now convert this to a TensorRT Directory**

c. Convert the .savedmodel file to a TensorRT directory by running this code:

FOR PYTHON 3.8

python

>>from tensorflow.python.compiler.tensorrt import trt\_convert as trt

>>import os

>>saved\_model\_path = “models/yourmodel\_converted.savedmodel”

>>tensorrt\_model\_path = “models/yourmodel”

>>os.makedirs(tensorrt\_model\_path, exist\_ok=True)

>>converter = trt.TrtGraphConverterV2(

input\_saved\_model\_dir=saved\_model\_path,

precision\_mode=trt.TrtPrecisionMode.FP16

)

>>converter.convert()

>>converter.save(tensorrt\_model\_path)

FOR PYTHON 3.6

python

>>from tensorflow.python.compiler.tensorrt import trt\_convert as trt

>>import os

>>saved\_model\_path = “models/yourmodel\_converted.savedmodel”

>>tensorrt\_model\_path = “models/yourmodel”

>>os.makedirs(tensorrt\_model\_path, exist\_ok=True)

>>conversion\_params = trt.ConversionParams(precision\_mode="FP16")

>>converter = trt.TrtGraphConverterV2(

input\_saved\_model\_dir=saved\_model\_path,

conversion\_params=conversion\_params

)

>>converter.convert()

>>converter.save(tensorrt\_model\_path)

Now we have a tensorrt directory for Python 3.6 & 3.8 which we can run and test!

TROUBLESHOOTING:

When trying to use TensorRT models made in different versions of Python with different dependencies, there were errors running the model using manage.py. This error came from not being able to load the savedmodel path

OSError: SavedModel file does not exist at: ~/projects/d4/models/yourmodel\_tensorrt/{saved\_model.pbtxt|saved\_model.pb}

To check whether your saved\_model.pb will load run this code:

>>import tensorflow as tf

>>model = tf.saved\_model.load("/home/jetson/projects/d4/models/suarez\_tensorrt")

>>print(model.signatures)

If this fails, the model might not be correctly converted.

**2. Benchmarking w/ TensorRT**

To use profile.py with a directory, we must change how our manage.py loads a model.

In manage.py we change the load\_model\_json function and add under model\_reload\_cb = None:

model\_reload\_cb = None

**###**

**if os.path.isdir(model\_path):**

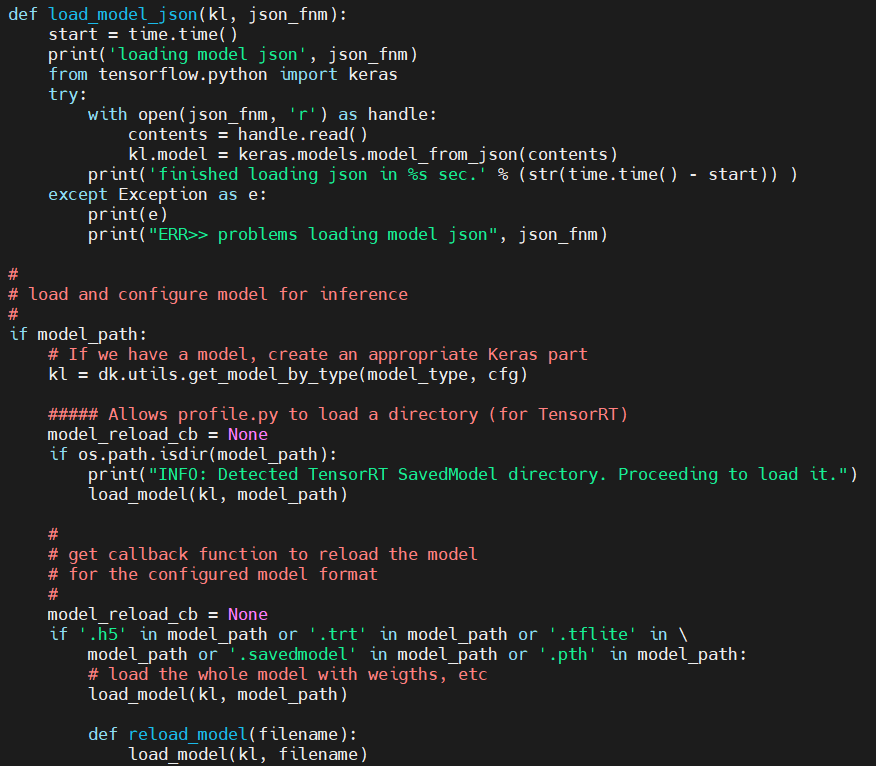
**print("INFO: Detected TensorRT SavedModel directory. Proceeding to load it.")**

**load\_model(kl, model\_path)**

###

elif ‘.h5’ in model\_path or ‘.trt’ in model\_path or ‘.tflite’ in \ model\_path or ‘.savedmodel’ in model\_path or ‘.pth’ in model\_path:

load\_model(kl, model\_path)



Once this manage.py file is saved, we can run profile.py using this command line:

python profile.py --model=./models/yourmodel\_tensorrt --type=tensorrt\_linear

**Drive Model:**

python manage.py drive --model=./models/yourmodel\_tensorrt --type=tensorrt\_linear