**Deploy Deep Learning Model using Graphpipe**

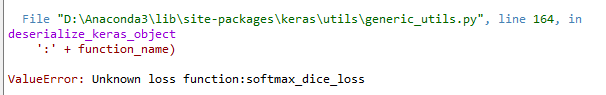
Over the past few years, there is a pretty progress in machine learning. Todays, you can grab one of handful frameworks like Keras, Tensorflow and pyTourch to generate machine learning model within hours.

But, the problem is there are many challenges to deploy the trained model into production environment. Here, I will explore how to deploy the model with few lines of coding using GraphPipe and predominantly get rid out from headache.

There is a proven metrics post about [GraphPipe](https://blogs.oracle.com/developers/introducing-graphpipe). You can find more about it with documentation, examples here <https://oracle.github.io/graphpipe>.

Here, I will demonstrate to deploy a trained model using keras API and segregate the deployment method into two.

1. With custom objects (like custom loss function, metrics, etc.,):
   1. save the model as json
   2. save the model weights as .h5
   3. again save the entire model as .h5
   4. now, you can follow the <https://github.com/oracle/graphpipe-tf-py/blob/master/examples/RemoteModelWithGraphPipe.ipynb> to convert the keras model into tensorflow .pb graph file
   5. You should do the step a to c. So, when loading the model, the all custom objects will able to load with NO error. Else will get an error message as shown below since the loss function which I used for training is custom one and it is not in the Keras utils

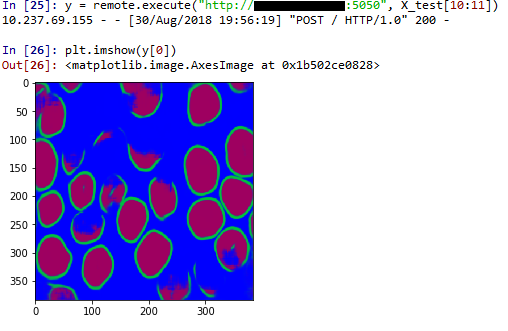


* 1. There is an alternate way to mention the custom object. But, above the way is very easy and no need to declare the custom function in the production environment

<https://github.com/keras-team/keras/issues/3977#issuecomment-289036668>

1. With No custom objects:
   1. Simply save the model as .h5
   2. Now, you can follow the <https://github.com/oracle/graphpipe-tf-py/blob/master/examples/RemoteModelWithGraphPipe.ipynb> to convert the keras model into tensorflow .pb graph file

GraphPipe Prediction:



Normal Prediction:

