

**Build an RNN (and then Some!)**

**Jean Bartik Symposium, February 1, 2019, Machine Learning Workshop Session**

**1530 – 1700**

*By members of the Booz Allen Women in Data Science Group and the Cakes & Tensors Booz Allen AI Meetup:*

*Catherine Ordun, Kristof Ladny and Michael Fagundo*

**Setting Up**

For this tutorial you will use a Python Jupyter Notebook file (.ipynb) loaded in the new Google Colaboratory environment that allows for 12 hours of free GPU run-time on a Tesla K80. For this session you don’t need anything more than your laptop and a Google account in order to access your Google Drive. Everything you need is located on this publicly accessible Google Drive Folder. Since this is connected to my own personal drive, please do not share this link outside of the tutorial. It will be shut down after the tutorial is finished.

Step 1- Log into your personal Google Drive account (like how you access your gmail). *You must be using Chrome, or you may encounter Google Colaboratory errors.*

Step 2 – Click on the link below to access the ‘Tutorial’ folder.

|  |
| --- |
| <https://drive.google.com/drive/folders/1jS6SJxAvxJQq035PbZA2tFh1fwuuY6mG?usp=sharing> |

Step 3 – Right click on the folder and “Add to your local drive”.

Step 4 – Now, right click on any “.ipynb” file. This is a Jupyter Notebook. When you right click, do you see “Colaboratory”?

Step 5a – If yes, go ahead and click it and the notebook will open inside Colaboratory for you.

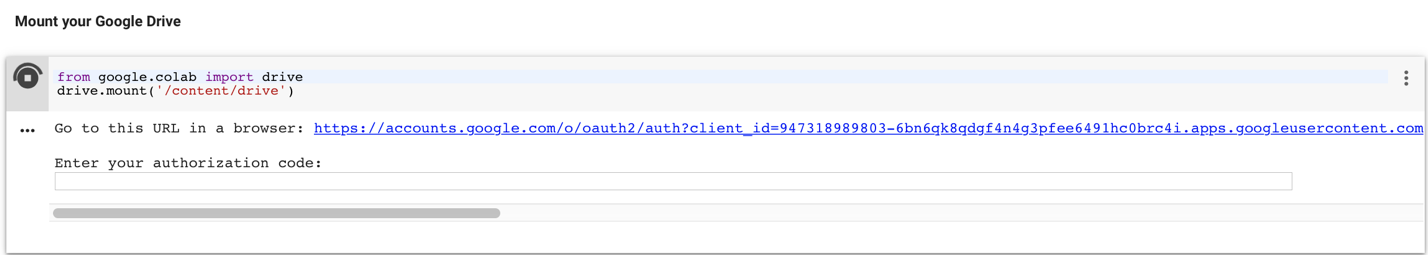
Step 5b – If not, continue to right click and “Connect more apps”

Step 6 – The Google apps frame will open, search for “Colaboratory” in the upper right hand corner search bar. Remember, it’s one ‘l’, not ‘ll’ (two).

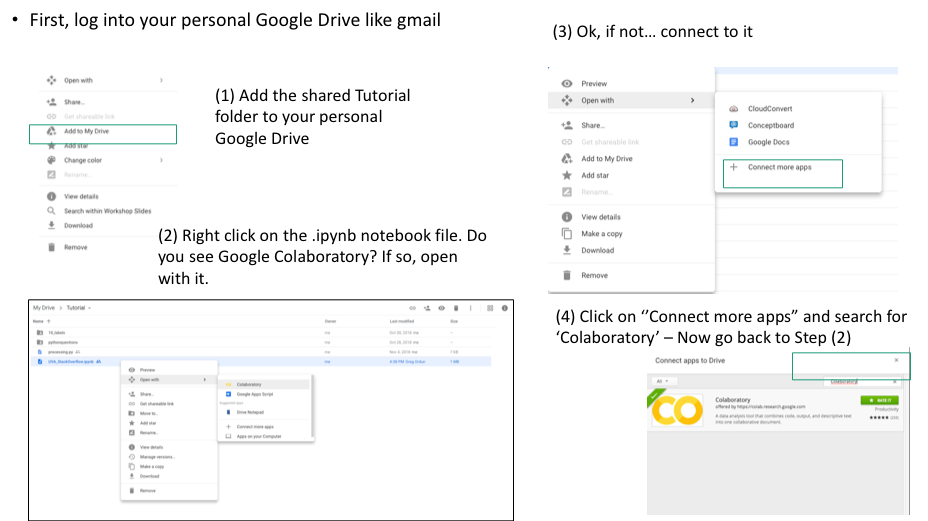
Step 7 – Follow the instructions and “connect” to it, next to the “share” icon. Give it a second to update.

Step 8 – Go back to Step 5a – right click on the “.ipynb” file and open in Colaboratory.

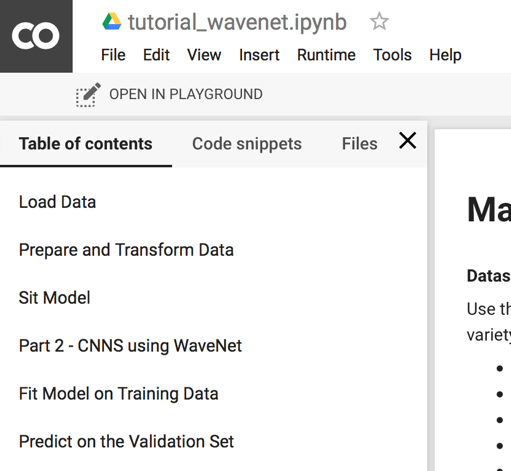
Step 9 – For this workshop, “Open in Playground”

When you get to the “Mount your Google Drive” cell, it will direct you to open a URL. Choose your Google Drive account, click approve, then copy and paste the string provided into the “Enter your authorization code” field. Now you’re ready!

*See below for additional screenshots:*



Open in Playground:

****

**Purpose of the “Build a RNN” Workshop**

This workshop is designed for students who are just learning Python and seeking more examples of intermediate code, to students who have dabbled in machine learning and are curious about deep learning models. No pre-requisites are required, just an interest to learn! We hope that you’ll be able to learn how to use a Jupyter Notebook, learn how to process data using Python, and also get some exposure to two important algorithms when dealing with time, sequence-based data.

**Introduction to Recurrent Neural Networks (RNN): ~ 45 minutes (Catherine & Michael)**

For the RNN and the WaveNet sessions, our main dataset will be the PAMAP2 Physical Activity Monitoring Data Set, focusing on one test subject’s sensor data.

In the RNN tutorial you’ll be learning how to build both a “vanilla” RNN using only the Python numpy library and a Long Short-Term Memory (LSTM) RNN in Keras. We will build out an RNN in numpy (see: **rnn\_numpy.ipynb**) to demonstrate the fundamentals of deep learning, tensors (otherwise called matrices) and operations on them. For the numpy RNN demonstration we will shift to using a song lyrics datatset to show how an RNN can be trained for character-by-character sequence generation.

With that fundamental understanding of basic RNN code, we will embark on building out an LSTM for time series prediction on the PAMAP2 dataset using the Keras library ((see: **tutorial\_rnn\_lstm.ipynb**). With the ability to store memory in an “internal state,” these networks are quite effective in predicting sequences, such as time series and text.

You need to first begin by opening the (see: **tutorial\_rnn\_lstm.ipynb**) file. We will then open the **rnn\_numpy.ipynb** file.

**Introduction to 1D Convolutional Nets Using WaveNet: ~ 20 minutes (Kristof)**

Recent research suggests that using convolutional neural networks (CNNs) are just as effective at sequence prediction tasks and have advantages in that they take less time to train and are more interpretable. For a “bake off” we will use the same PAMAP2 sensor data set but apply it on a one-dimensional (non-spatial) CNN using the basic version of a popular temporal convolutional network called “Wavenet” developed by Google as the primary machine language translation algorithm. Wavenet in its basic form can be implemented in Keras which is what we will do today (see: **tutorial\_wavnet.ipynb**). Let’s take a look at how Wavenet using a temporarl CNN works in comparison to the LSTM. We will also introduce new ways to process the sensor data to prepare it for the CNN.

**Presenters**

Catherine Ordun, Senior Data Scientist: [ordun\_catherine@bah.com](mailto:ordun_catherine@bah.com)

Kristof Ladny, Lead Data Scientist, 2008 Alumnus: [ladny\_kristof@bah.com](mailto:ladny_kristof@bah.com)

Michael Fagundo, Data Scientist, [fagundo\_michael@bah.com](mailto:fagundo_michael@bah.com)