

# User Guide to run the code for 'Interpreting the Filters in the First Layer of a Convolutional Neural Network for Sleep Stage Classification'

Gulrukh Turabee

## 1. Installation and Dependencies:

### 1.1 Anaconda

Anaconda current versions are available online and can be downloaded from <sup>1</sup>. The Anaconda version used for this project is **5.3.1** with Python **2.7** version.

### 1.2 Installation of Tensorflow and Keras on Anaconda

After installing Anaconda, a working environment needs to be setup. The step-by-step guide to create a working environment is as follows:

- 1) Open anaconda Prompt from the Start menu and run the following line of code to create a new environment as 'anaconda\_project1'. Note that the name of the environment may vary.

```
conda create -n anaconda_project1 python=3.5
```

```
(base) C:\Users\n0818006>conda create -n anaconda_project1 python=3.5
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\n0818006\AppData\Local\Continuum\anaconda2\envs\anaconda_project1

added / updated specs:
- python=3.5

The following NEW packages will be INSTALLED:

certifi          pkgs/main/win-64::certifi-2018.8.24-py35_1
pip              pkgs/main/win-64::pip-10.0.1-py35_0
python           pkgs/main/win-64::python-3.5.6-he025d50_0
setuptools       pkgs/main/win-64::setuptools-40.2.0-py35_0
vc               pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime  pkgs/main/win-64::vs2015_runtime-14.15.26706-h3a45250_4
wheel            pkgs/main/win-64::wheel-0.31.1-py35_0
wincertstore     pkgs/main/win-64::wincertstore-0.2-py35hfdbdb8_0

Proceed ([y]/n)?
```

- 2) Press 'Y' and let the new packages get installed. Once everything is installed, it will ask to "Activate it".
- 3) In order to activate the environment, execute the following command. Once the user runs the above command, prompt will change. It should look like below image.

---

<sup>1</sup> <https://www.anaconda.com/distribution/>

```
activate anaconda_project1
```

```
(base) C:\Users\n0818006>activate anaconda_project1  
(anaconda_project1) C:\Users\n0818006>
```

- 4) This means that now the user is inside this environment and can install any package independently. Now the user needs to install Tensorflow, Keras, and Spyder in this environment.

```
pip install --ignore-installed --upgrade tensorflow
```

```
(anaconda_project1) C:\Users\n0818006>pip install --ignore-installed --upgrade tensorflow  
Collecting tensorflow  
  Downloading https://files.pythonhosted.org/packages/c8/58/531ff043b41dd9df78aad7fca4adc57fb6a43bee4262c9419cccf16e2c1/tensorflow-1.14.0-cp35-cp35m-win_amd64.whl (68.3MB)  
100% |#####| 68.3MB 231kB/s
```

- 5) After this, install Keras version 2.2.4 by typing the following command using pip. Pip is a package manager for python packages.

```
pip install keras==2.2.4
```

- 6) Now install HDF5 storage using pip. This package provides support to read/write HDF5 formatted files, which includes MATLAB (.mat) v7.3 formatted files as well.

```
pip install hdf5storage=0.1.15
```

- 7) Next, install Matplotlib using pip to save the file as MATLAB (.mat) v7.3 files.

```
pip install matplotlib
```

- 8) Then install Scikit-learn library using pip. This library features various algorithms.

```
pip install matplotlib
```

- 9) Finally, install Spyder using the following command.

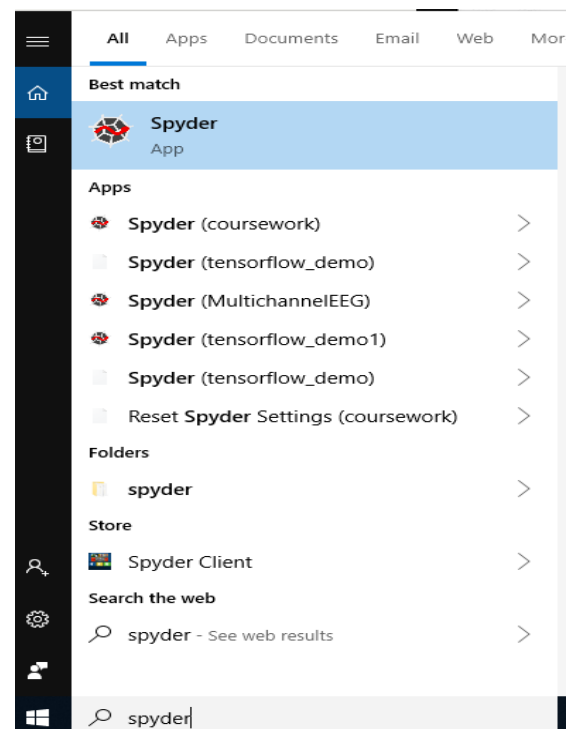
```
conda install spyder
```

- 10) If the user wishes to delete the current environment and make a new one, this can be done by running the following command.

```
conda env remove -n anaconda_project1
```

- 11) After installing all the required packages, go to start menu and search for Spyder IDE (note that for each environment a different Spyder is installed, open the recently installed Spyder with environment name in the bracket as shown in image on the left) or user can also load Spyder by simply typing it on the command prompt as shown in an image on right.

```
(anaconda_project1) C:\Users\n0818006>spyder
```



### 1.3 Software Dependencies

In order to run this code user needs to install the Numpy and Scipy packages as well which gets automatically installed while installing TensorFlow, however, if not they can be installed using pip like this:

```
pip install Numpy
```

```
pip install scipy
```

## 2. Data Collection and Pre-processing: Sleep EDF Database

Download the raw data PSG and the corresponding hypnogram file ({subject}. PSG.edf and {subject}. Hypnogram.edf) from the sleep EDF website <sup>2</sup>. Note that for this project, the first 39 subjects have been used from the given dataset.

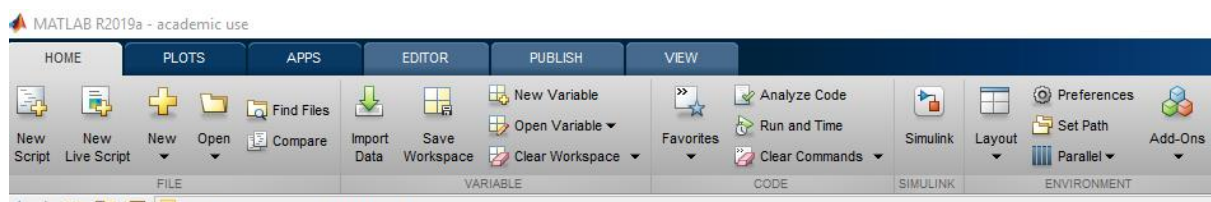
## 2.1 Reading Sleep .edf files in Matlab Software

The raw data is in the form of .edf (European data format). In order to read these files in MATLAB software, a file reader is needed. edfRead is a simple file reader to .edf files in MATLAB. Moreover, edfreadUntilDone is the modification of edfread; it is meant to overcome file size limitations of the former. Download these two files ('edfread.m' and 'edfreadUntilDone.m') inside the current working directory.

## 2.2 Set path in MATLAB Software

Before executing any piece of code, make a directory named as '**Data**' inside the current working directory and set the path of the current working directory inside MATLAB. This can be done by following the steps below:

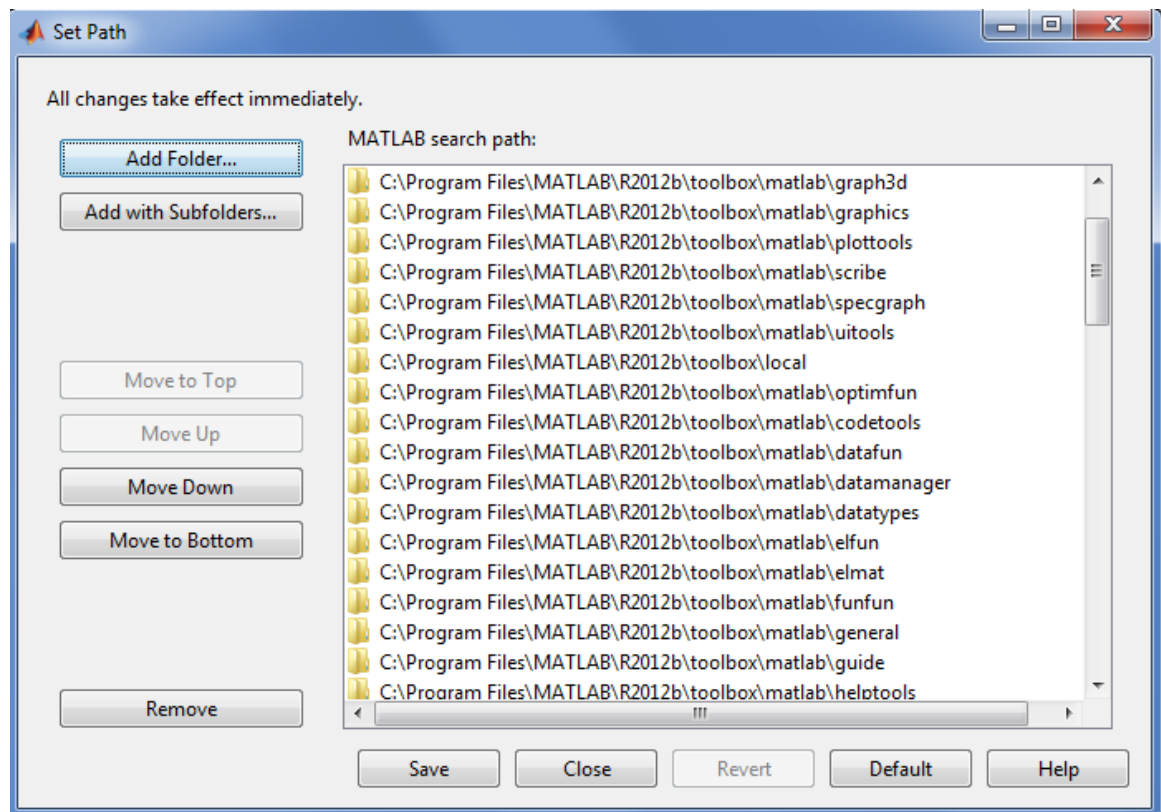
- 1) Download and open MATLAB Software (any version). Locate the option of set path in the 'Home' Tab.



---

<sup>2</sup> <https://physionet.org/content/sleep-edf/1.0.0/>

- 2) Click on set path, and then click on 'Add Folder...' Option and select the current working directory.



- 3) Click on save and now the user is all set up to execute the code in MATLAB.

### 2.3 Saving .edf files as MATLAB (.mat) File for all 39 Subjects

Now it is time to compile all .edf files into one .mat file. The code for this is also available at GitHub repository file namely as '**collect\_all\_data.m**'. Instructions on how to run the code are written inside the .m file.
















### 2.4 Extract Data as Individual MATLAB (.mat) files for all 39 Subjects

Make a directory named as 'Data'.

Run '**datastruct.mat**' file to create the 39 MATLAB (.mat) data files with each subject as testing data with file name as DataTestSub {subjectIndex}.mat.

Since there are 39 subjects, using leave-one-subject out cross-validation, implementing this code will make 39 MATLAB (.mat) files with each subject as testing data with the filename as DataTestSub\_{subjectIndex}.mat. The training data is stored as cell arrays, X\_train and Y\_train, where each cell contains a matrix. The testing data, X\_test and Y\_test are stored as matrices.

Copy these files into the 'Data' folder. The files will look like this as shown below:

Name	Date modified	Type	Size
 DataTestSub_1.mat	17/12/2018 17:31	MATLAB Data	867,486 KB
 DataTestSub_2.mat	12/12/2018 00:34	MATLAB Data	867,486 KB
 DataTestSub_3.mat	17/12/2018 17:34	MATLAB Data	867,486 KB
 DataTestSub_4.mat	17/12/2018 17:35	MATLAB Data	894,621 KB
 DataTestSub_5.mat	17/12/2018 17:36	MATLAB Data	867,486 KB
 DataTestSub_6.mat	17/12/2018 17:37	MATLAB Data	867,484 KB
 DataTestSub_7.mat	17/12/2018 17:38	MATLAB Data	867,486 KB
 DataTestSub_8.mat	17/12/2018 17:40	MATLAB Data	867,484 KB
 DataTestSub_9.mat	17/12/2018 17:41	MATLAB Data	867,484 KB
 DataTestSub_10.mat	17/12/2018 17:42	MATLAB Data	867,486 KB
 DataTestSub_11.mat	17/12/2018 17:43	MATLAB Data	867,486 KB
 DataTestSub_12.mat	17/12/2018 17:44	MATLAB Data	867,484 KB
 DataTestSub_13.mat	17/12/2018 17:46	MATLAB Data	867,484 KB
 DataTestSub_14.mat	17/12/2018 17:47	MATLAB Data	867,486 KB
 DataTestSub_15.mat	17/12/2018 17:48	MATLAB Data	867,484 KB

User can load any of the files in the Matlab, using the following command:

```
a = load ('C:\Users\n0818006\Desktop\collect_data_code\DataTestSub_1.mat')
```

As a result, the user will see a Matlab structure with following fields:

```
a =  
  
  struct with fields:  
  
    TestSubIdx: 1  
      X_test: [2650×3000 double]  
    X_train: {36×1 cell}  
      Y_test: [2650×1 double]  
    Y_train: {36×1 cell}
```

Now that we have all the data files ready, it is time to do training of the data using the DeepSleepNet architecture.

### 3. DeepSleepNet Model Architecture

Make the following directories inside the current working directory:

('../preTrainResults')

('../results/ supPreTrain+Test')

('../activations/activation\_sf')

Open Spyder using the Anaconda command prompt and run the following python (.py) files step by step to train and test the data:

- supPreTrainDeepSleepNet.py
- supPreTrain+TestDeepSleepNet.py

Then, run 'activation\_sf.py' file to extract activations and save activations as .mat file from 1st CNN layer.

### 3. Selection of Sleep Stage 2 EEG epochs with prominent EEG Patterns via Welch PSD

For each subject, we are to select five 1-minute EEG epochs that best exhibit for the prototypical EEG patterns occurring in sleep stage 2, the algorithm to perform this task was implemented in MATLAB. It first computed Welch Power Spectrum Density for all 1-minute EEG epochs from the recording. Although there are 39 subjects in the cohort, however, only 34 EEG recordings are used as remaining did not have sleep stage 2. The 34 EEG recordings .mat files have been used as an input and the predicted values saved as 'predictionTestSub'+Subject\_Index+'.mat' files have been used to input only those EEG patterns which have sleep stage predicted values as 2.

Run 'Sleep\_stage\_2\_welch\_PSD.m' file to find the top 5 Sleep Stage 2 EEG epochs with prominent EEG Patterns via Welch PSD.

### 4. Top Ranked Kernels (Sleep Stage 2):

Now that we have the top five significant epochs for each subject, we can estimate the top ranked kernels among those selected epochs. For that run the 'Sleep\_stage\_2\_top\_ranked\_kernels.m' file.

### 5. Yule Walker Method to computer PSD of Top ranked Filters (Sleep Stage 2):

After obtaining the top ranked kernels of all subjects for sleep stage 2 epoch, the next step is to compute the power spectral density estimate of learned CNN kernels. For that run 'Sleep\_stage\_2\_Yule\_Walker.m' to plot and compute PSD of learned kernels.

Similarly, run 'Sleep\_stage\_4\_welch\_PSD.m', 'Sleep\_stage\_4\_top\_ranked\_kernels.m' and 'Sleep\_stage\_4\_Yule\_Walker.m' to get the results for sleep stage 4.

**Please cite: Cite this paper as:**

Turabee G., Shen Y., Cosma G. (2020) Interpreting the Filters in the First Layer of a Convolutional Neural Network for Sleep Stage Classification. In: Ju Z., Yang L., Yang C., Gegov A., Zhou D. (eds) Advances in Computational Intelligence Systems. UKCI 2019. Advances in Intelligent Systems and Computing, vol 1043. Springer, Cham