Hybrid Deep Learning Model for Exposure Classification Based on Rodent Electrophysiology

Max Copeland

Master of Computer Science Pursuant, University of Montana, Missoula, USA

E-mail: maximillian.copeland@umontana.edu

Abstract

This study proposes a hybrid deep learning architecture for classification of electrophysiology signal data collected from rodent socialization experiments. The purpose is have a model that can predict types of exposures (exposed to another rodent, a wooden block, etc) from time dependent frequency channels. The proposed model leverages feature extraction capabilities while also supporting sequential nature of time dependent input data, resulting in faster model training for exposure classification without significant loss in accuracy.

Keywords: neuroscience, deep learning, electrophysiology

1. Source Data Background

Rodents have long served as effective test vehicles for explorations in behavioral neuroscience. Though breeds of mice and rats tend to be standard for the field, an exceptional subject for studies in behavior neuroscience is the Degu, a breed of rodent who possess exceptional social behavior--such as exhibiting complex colony decorum and a vocalized language, among others.

As such, the source data in this paper is generated from lab experiments surrounding Degu social behavior. The experiments are conducted as follows: neurosurgery is performed on the degu subject, implanting roughly twelve tetrodes, localized around both the pre-frontal cortex and hippocampus. These tetrodes feed the local field potential voltage to hardware that writes this signal data to files, (one file per tetrode) for later analysis. After sugergy, the degu subject is given appropriate recovery time. Experimentation begins with the subject degu being placed in a small pen, and given lights-off rest time. At the conclusion of rest time, a series of exposures are performed, each lasting on the order of minutes. These exposures include placing a wooden block in the pen, placing a familiar degu in the pen, placing a 3D printed degu object in the pen, to name a few. The experiment is concluded by another round of lights-off rest time.

The scope of this project is to analyze the local field potential data for hippocampal gamma oscillations, which have been found to indicate the access of short-term memory [1]. Further, the aim of this project is to build a machine learning model that can correctly classify exposure types based on gamma-filtered local field potential data from the hippocampus and pre-frontal cortex. The aim of this model is to tell us what exposures the subject degu is using its memories to govern its interaction.

The total breadth of experimentation spans across four degus, each seeing dozens of experiments, and with each experiment generating data on roughly a dozen channels amounting to a volume of data on the order of terabytes. For the purposes of this project, data from only one degu experiment is used as a proof of concept, which will be considered to generalize across the full repository of data.

1.1 Data Preparation and Cleansing

A significant overhead to training a machine learning model to the raw channel electrophysiology data was to effectively read, parse, and munge the source data into a format suitable for interfacing via a programming language (in this case, Python). Source data was collected in lab experiments utilizating a proprietary sensor hardware to collect local field potentials. As such, the source data files are written to a proprietary file format. DegPy is a custom package I built specifically for the reading, parsing, and processing of this raw, cryptic data.

The Degpy module utilizes an overarching object for interfacing with the data called `Session`. Through this object, we can access metadata at the experiment level, as well as access data at the Terminal level via the `.get_terminal()` method. This method returns a `Terminal` object, which provides access to data collected from an individual tetrode. Through these objects, we can read in all of the raw channel data into a numpy array for downstream processing, namely gamma filtering a normalization, per the scope of the project (Fig.1, Fig2).

In the `Terminal` object exists a 1D array of all exposures, which is our target variable. However, the raw target is a sub-optimal format for training. The raw target contains labels such as `'b1s'`, which indicates a wooden block has been inserted into the Degu pen, while `'b1e'` indicates the block has been removed. Additionally, there are combinations of exposures, e.g. `'s0b1s` denotes another degu being added to the pen in addition to a wooden block. As such, this format does not lend itself to a simple one-hot matrix offered by popular ML packges.

To enable a representative target matrix, I created a custom method on the `Terminal` object that parses the labeling convention and converts it into a multi-hot target matrix to indicate which exposures are "on" or "off".

In short, this module delivers all upstream data processing for model training (some preprocessing steps are rewritten in the corresponding Jupyter notebook for completeness).

There are two notes to make from a data perspective: 1) the size of data from a single experiment is prohibitively large for training locally. As such, I made the decision to evenly downsample the data to be 1% of the source data size. This decision likely sacrifices completeness of the model, however, the comparitive accuracies of the models presented should still persist. 2) A standard methodology in the field for feature decomposition is to use Principal Component Analysis to narrow the space of input data, which is a step I deliberately avoided. One aim of this project is to see if implementating a Convolutional layer can serve this same purpose.

1.1.1 Model Selection and Training. The nature of our source data poses an interesting situation, in which we have a classification problem, but the time dependence of the data means that the order of the datapoints matters.

It is for these reasons that I chose the Long Short-Term Memory Recursive Neural Network model. An important feature of RNN's is that there is a sense of "state" at each time step of the classification-- a property that lends itself well to the source data. However, RNN's cannot bridge more than 5-10 time steps-- a problem solved by LSTM models [2]. However, being that I deliberately neglect to do any feature decomposition, the aim is to have this done by a convolutional layer from which the LSTM layer is fed. The goal is faster training time without loss of accuracy.

For a control, a base LSTM model is trained, consisting of a 100 unit LSTM layer, a dropout layer, and a softmax activated fully connected layer. This model is compared to the proposed model, which is comprised a 2D CNN layer, LSTM layer, a dropout layer, and a softmax-activated fully connected output layer. The 2D CNN layer and LSTM layer are conveniently abstracted by Keras in the ConvLSTM2D layer object.

The downsampled input data was split into training and test sets (proportions of 75% and 25% respectively).

Model Evaluation

Through ten epochs of training, the base model and hybrid model performed with accuracies of 93.3% and 93.4% respectively on the withheld test data (Fig.4), thus adding a convolutional layer preceding the LSTM layer results in no loss in accuracy of the model. The base model and hybrid model training times executed in a median of 0.7685 and 0.391 milliseconds. per sample respectively (Fig.3). This is a 58% improvement in training time without any loss in classification accuracy. This hybrid architecture, while not novel, demonstrates extensibility to the full repository of data to significantly improve training time where a strictly LSTM model training time may not be practical.

References

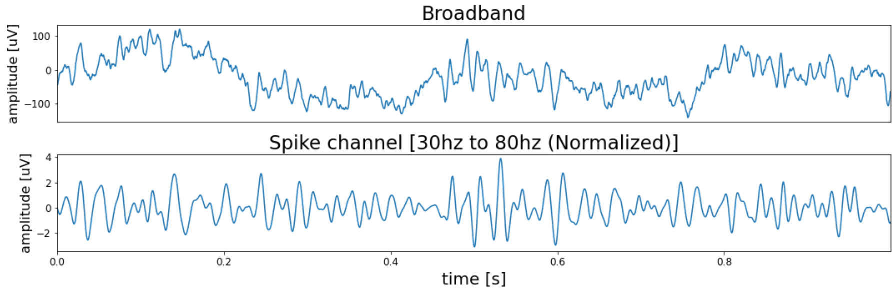
[1] van Vugt, Marieke K et al. "Hippocampal gamma oscillations increase with memory load." *The Journal of neuroscience : the official journal of the Society for Neuroscience* vol. 30,7 (2010): 2694-9. doi:10.1523/JNEUROSCI.0567-09.2010

[2] Staudemeyer, Ralf & Morris, Eric. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks.

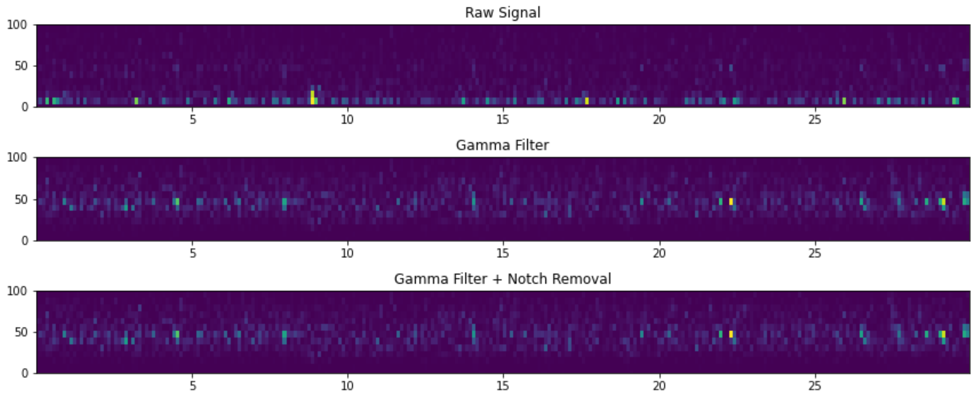
[3] A. Supratak, H. Dong, C. Wu and Y. Guo, "DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998-2008, Nov. 2017, doi: 10.1109/TNSRE.2017.2721116.

Figures

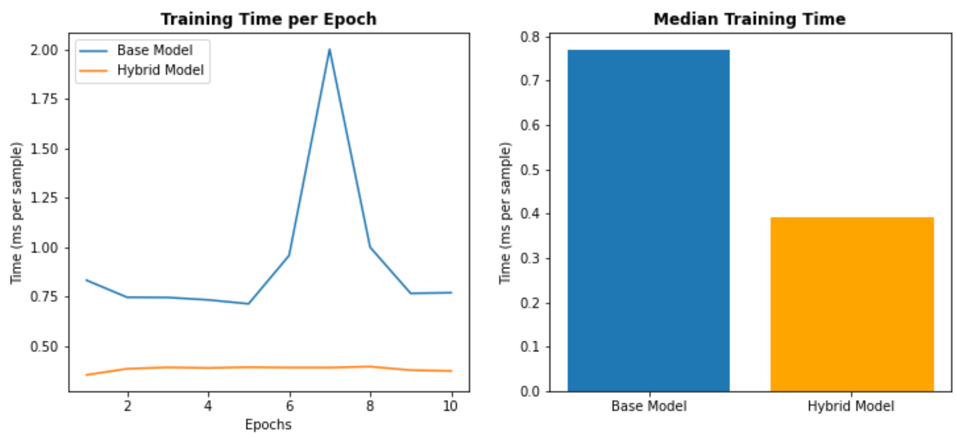
[Fig.1]



[Fig.2]



[Fig.3]



[Fig.4]