Steinmetz Dataset Research Plan

Project Overview

Abstract:

- **Background:** Motor control involves complex and dynamic interactions across multiple cortical regions, however current neural decoding approaches often treat neural activity as static patterns or simple sequences. To address this, we aim to develop a deep learning framework that decodes movement intentions from large-scale neural recordings in the primary(MOp) and secondary (MOs) motor cortex of a mouse, with emphasis on capturing behaviorally relevant temporal structure.
- **Methods:** We utilized the Steinmetz et al. 2019 dataset and extracted neural spike train data recorded from MOp and MOs regions of one mouse during a two-alternative unforced choice task. Spikes were binned at 10 ms resolution and aligned to key behavioral events, including the go-cue, wheel movements, and response feedback. The dataset was split by session into training, validation, and test sets to prevent data leakage across. We use three long short-term memory (LSTM) architectures sequentially to decode left/right turn choice decisions from motor cortex neural dynamics, centered around go-cue, movement, and feedback events. Additionally we retain hidden state data in order to cross-correlate if the intermediate representation of wheel position movement correlates to the ground truth (wheel velocity generated from wheel position data from the dataset). Finally we compare the final output to the expected trial outcome.

  Conclusion: This study provides a computational framework for decoding dynamic motor behavior from cortical activity in the mouse brain using deep recurrent networks. Beyond decoding performance, this approach allows investigation of the functional roles of different brain regions in the dynamic behavioral process involved in decision making. Using a bottom up addition approach or vice versa, one could generalize the neuronal group structures by cross-correlating the hidden states with the wheel position at each time step across different contrasting stimuli on either side.

Primary Goal: Develop a foundation for transitioning from conventional deep learning to biologically plausible spiking neural networks using real neural data.

Dataset Background

Steinmetz et al. 2019 Study

- **Paper**: "Distributed coding of choice, action and engagement across the mouse brain" https://drive.google.com/file/d/1whffUVCcF-656XudydRr3QbeoLvi49LG/view?usp=drive_link
- **Published**: Nature, 2019
- **Significance**: Largest simultaneous multi-area neural recording dataset at the time
- **Access**: Available through Allen Institute Brain Observatory

Experimental Setup

Task: Visual discrimination task with motor response

- Mice view visual stimuli (gratings) on left and/or right screens
- Different contrast levels (0%, 25%, 50%, 100%)
- Mice turn steering wheel left or right based on highest contrast
- Reward given for correct choices
- Trial structure: gocue (0.4s) → wheel movement epochs (up to 60s) → feedback

Recording Technology:

- **Neuropixels probes**: High-density silicon probes (Neuropixels 1.0)
- **384 recording sites** per probe, up to 3 probes simultaneously
- **30 kHz sampling rate** for high temporal resolution
- **Chronic implants**: Multiple sessions per animal

Dataset Specifications

Neural Data Structure

- `spikes.times`: Spike timestamps (seconds)
- `spikes.clusters`: Cluster IDs (cluster index for that session) for each spike
- `spikes.amps`: Spike amplitudes
- `spikes.depths`: Electrode depth positions

Cluster Information:

- `clusters._phy_annotation`: 0 = noise (these are already excluded and don't appear in this dataset at all); 1 = MUA (i.e. presumed to contain spikes from multiple neurons; these are not analyzed in any analyses in the paper); 2 = Good (manually labeled); 3 = Unsorted. In this dataset 'Good' was applied in a few but not all datasets to included neurons, so in general the neurons with _phy_annotation>=2 are the ones that should be included.

- `clusters.depths`: Anatomical depthThe position of the center of mass of the template of the cluster, relative to the probe. The deepest channel on the probe is depth=0, and the most superficial is depth=3820.
- `clusters.peak_channel`: The channel number of the location of the peak of the cluster's waveform.
- `clusters.probes`: The probe on which the cluster was detected.
- `templateWaveformChans`: The indices of the top 50 channels for this neuron's waveform, by amplitude.
- `templateWaveforms`: The template waveform shapes (across 82 time samples at 30kHz) on the top 50 channels, by amplitude. This dataset is to be considered together with templateWaveformChans. From the two of these, you can construct a full matrix of size (nClusters,82,384) of the template shapes across all channels.

Behavioral Data

Trial Structure:

- `trials.feedbackType:` -1 for negative feedback (white noise burst); +1 for positive feedback (water reward delivery`).`
- `trials.feedback_times:`Feedback timestamps
- `trials.goCue_times:` The 'goCue' is referred to as the 'auditory tone cue' in the manuscript.
- `trials.repNum:` Trials are repeated if they are "easy" trials (high contrast stimuli with large difference between the two sides, or the blank screen condition) and this keeps track of how many times the current trial's condition has been repeated.
- `trials.response_choice:` The response registered at the end of the trial, which determines the feedback according to the contrast condition. Note that in a small percentage of cases (~4%, see manuscript Methods) the initial wheel turn was in the opposite direction. -1 for Right choice (i.e. correct when stimuli are on the right); +1 for left choice; 0 for Nogo choice.
- `trials.response_times:` Response Timestamps
- `trials.intervals:` start-stop times of trial intervals

Stimulus Information:

- `trials.visualStim_contrastLeft:` A value of 0.5 means 50% contrast. 0 is a blank screen: no change to any pixel values on that side (completely undetectable).
- `trials.visualStim_contrastRight:` Contrast on the right`.`
- `trials.visualStim_positions`: Stimulus positions

Movement Data:

- `wheel.position`: The wheel has radius 31 mm and 1440 ticks per revolution, so multiply by 2*pi*r/tpr=0.135 to convert to millimeters. Positive velocity (increasing numbers) correspond to clockwise turns (if looking at the wheel from behind the mouse), i.e. turns that are in the correct direction for stimuli presented to the left. Likewise negative velocity corresponds to right choices.
- `wheel.timestamps`: wheel position timestamps

Brain Regions Covered

Motor Areas (Focus for movement prediction):

- **MOp**: Primary motor cortex
- **MOs**: Secondary motor cortex

Research Question Details

Primary Hypothesis

Success Metrics

- **Prediction Accuracy**: correlation between predicted and actual wheel position
- **Temporal Precision**: Performance at different time bins (10ms, 1ms)

Technical Implementation Plan

Phase 1: Data Processing Pipeline

Input Preparation:

- Extract trials with MOs, MOp brain regions
- Bin neural spikes (multiple resolutions: 10ms, 1ms)
- Align neural and behavioral data timestamps
- Group neurons by response dynamics, then average them over trials
- Create train/validation/test splits by session

- Limit to one mouse (Cori)

Phase 2: Baseline Models

- LSTM to Signed Regression or Classification (try this first!) Head Output
- Softmax Outer Layer to predict trial success rate

Phase 3:

- Merge each unit, dataloaders, model, visualization into one pipelined process for scalability

---

Dataset Access & Tools

Data Access

- **Download link**: https://figshare.com/articles/dataset/Dataset_from_Steinmetz_et_al_2019/9598406
- **File format**: .npy files
- **Download size**: 8.25 GB (compressed), 15 GB (uncompressed)
- **Documentation**: https://github.com/nsteinme/steinmetz-et-al-2019/wiki/data-files\
- **Schema:** GoogleDrive link
- **Notebook:** GoogleDrive Jupyter Notebook

Required Tools

Data Processing:

- `numpy`, `pandas`: Data manipulation
- `scipy`: Signal processing

Machine Learning:

- `pytorch`/`tensorflow`: Conventional deep learning
- `sklearn`: Baseline models and metrics

Visualization:

- `matplotlib`, `seaborn`: Standard plotting
- `plotly`: Interactive visualizations

Expected Challenges & Solutions

Technical Challenges

1. **Large dataset size**: Use data streaming and batch processing
2. **Temporal alignment**: Careful timestamp synchronization
3. **Computational requirements**: GPU acceleration for training

Methodological Challenges

1. **Baseline comparison**: Ensure fair comparison between conventional and SNN approaches
2. **Hyperparameter tuning**: Systematic optimization for both model types
3. **Overfitting**: Cross-validation with session-wise splits
4. **Interpretability**: Develop methods to understand learned representations

---

Timeline & Milestones

Week 1-2: Hypothesis Research and Task Management

- ☑ ~~**Iyad Ba Gari,** repo management, upload the `requirements.txt` or setup.py file~~
- ☑ ~~**Keming Liu,** play with toy dataset, center spike counts to events~~
- ☐ **Isaac Thu,** technical background research on dynamical deep learning models for motion prediction, deployment
- ☐ **Abu Mohammed,** hypothesis research and finalize dataset loader functions

---

Week 3:

- ☐ **Abu Mohammed,** Finalize neuron dynamics grouping:
  - ☑ ~~Gocue: Standard Peak Detection~~
  - ☐ Wheel Movement: Rolling average, change point detection
  - ☐ Feedback: Peak detection, sustained firing
- ☐ **Isaac Thu:** LSTM model adapted to neuron spikes input and wheel direction/speed output

- □ **Iyad Ba Gari:** Merge individual units into a pipeline
- □ **Keming Liu:** Abstract and powerpoint presentation (collaborate with other team members)

Literature References

Key Papers

1. **Steinmetz et al. (2019)**: "Distributed coding of choice, action and engagement across the mouse brain" - Nature

---

Next Steps

1. **Immediate Actions**:
   - Set up development environment and data access
   - Begin exploratory data analysis
   - Implement basic preprocessing pipeline
2. **Team Coordination**
   - Daily progress meetings
   - Shared code repository setup
   - Division of implementation tasks

This document serves as a living reference and will be updated as the project progresses. Please add comments and suggestions for improvements.