MACHINE LEARNING TECHNIQUES FOR PREDICTING MOBILITY-RELATED PERCEPTION ERRORS IN ASTRONAUTS

Steven Belcher
Undergraduate Student
Computer Science/Mathematics Major

Faculty Advisor: Raj Dasgupta
Astronauts returning from long durations in outer space face problems while performing normal mobility–related functions

Neuro–vestibular problems:
- Inability to correctly perceive ambient distance and space,
- Incorrectly timing arm and leg movements
- **Balance disorder:** Inability to maintain posture and balance

Image source: National Space Biomedical Research Institute (NSBRI) and NASA
https://www.sciencedaily.com/images/2009/12/091209134640_1_540x360.jpg
Proposed Solution: Modular Robotics Suit (MORS)

- Attached to leg and arm joints of a person
- Automatically detect space and time perception errors during movement
Research Question

- Can machine learning (ML) techniques detect mobility related perception errors?

Proposed Approach

- ML algorithm analyzes time-series mobility data
  - collected from humans performing simple mobility tasks such as flexing arms or legs
- Automatically predict space-time perception errors, if any
- Work in real-time
Two ML techniques used
- Supervised Learning
- Semi-supervised Learning

Use deep neural network
- Builds a virtual model of the network of neurons similar to human brain
- Algorithm *shown* examples called training set
  - Format of example: Data Pattern (or Feature) → Label
- Weights of neural network edges adjusts as the algorithm learns
Two types of supervised learning algorithms
- Convolutional Neural Network
- Support Vector Machine
Convolutional Neural Network (CNN)

- The CNN being used is comprised of a series of Convolution and Max Pooling Layers, followed by a fully-connected output layer
  - The network input is a set of normalized signal vectors
  - Convolution layers use ReLu activation function
  - Output layer makes binary predictions indicating the presence or absence of an imperception
Support Vector Machine (SVM)

- The SVM is a simple linear classifier that is given a 130-dimensional feature set as input, generated by tsfresh\(^1\) package
  - As with the CNN, the SVM makes binary predictions, but uses a determined separating hyperplane to make assessments
  - This model is much simpler to build, but requires meaningful features be generated first
    - This adds to preprocessing time and can require additional domain knowledge to do well

---

Accuracy and Loss graphs for the data sets used in training the CNN are shown below. Loss thus far has had high variance, likely due to the relatively small size of the data sets generated thus far.

Using only raw signals, the CNN obtained an average test set accuracy of 88.98%

This indicated plenty of room for improvement through hyperparameter tuning and architecture alterations.
CNN vs SVM

Below we see a comparison between the CNN and SVM using the data gathered thus far.
- Even given limited, noisy data the CNN outperforms the SVM on test samples, though both perform decently
Problems with Supervised Learning

- Requires **large amounts** of **correctly labeled** training data
- Difficult to obtain in real-life
  - Noise while sampling data
  - Missing data
  - Mis-labeled data
  - Human errors in labeling
- **Solution:** Semi-supervised learning – learning with incorrect, missing, mis-labeled data
To deal with the issues of noisy labeling and small data sets moving forward, a self-ensembling technique called **Temporal Ensembling**\(^{[2]}\)

- Uses the outputs from the network-in-training at different epochs and under different input conditions to form a consensus about what the true labels are.
- Allows for the use of fuzzy or unlabeled data in training produced by data augmentation techniques.

---

Gathering more data is key moving forward, as this will allow for better accuracy and lower loss in the CNN.

Data augmentation techniques such as signal distortion and DBA\[^3\] are being used to generate synthetic data.

The use of Temporal Ensembling should enable more effective use of the noisy and unlabeled data, yielding higher accuracy and better generalization.

---

Acknowledgements

- NASA EPSCoR support for MORS project

Project Collaborators:
- Dr. Carl Nelson, University of Nebraska–Lincoln (MORS robot design)
- NASA Project Monitor: Dr. Jeffrey Bloomberg, Johnson Space Center
- Dr. Mukul Mukherjee, University of Nebraska–Omaha Biomechanics
- Kyle Brozek, University of Nebraska–Omaha Biomechanics Graduate Student