Embedded Neural Network to Detect Motion Activities

Samson Connelly and Kellan La Belle
Luddy School of Informatics, Computing, and Engineering
authors: sambconn@iu.edu, kllabell@iu.edu

Introduction

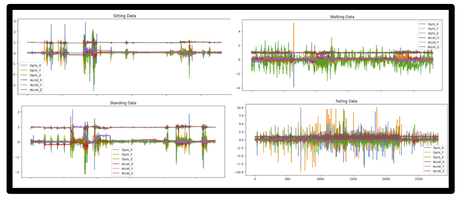
- **Service Overview:**
  - Detecting if someone is sitting, standing, walking, or falling
  - Falling is a leading cause of injury in elderly communities
  - Our service offers automatic detection that allows for timely assistance
- **Dataset:**
  - Accelerometer and Gyroscope data collected while performing all four actions
  - 6 Total Features:
    - Gyroscope X, Y, and Z
    - Accelerometer X, Y, and Z
  - 11,200 datapoints
- **Model Overview:**
  - We used a convolutional neural network to make our predictions
  - CNN's excel at recognizing patterns and extracting information from a complex dataset
  - Used a sliding window in order to make our predictions on more than just one data point

Objective

Develop a neural network that can accurately detect human motion activities and deploy it on an embedded edge device.

Methods

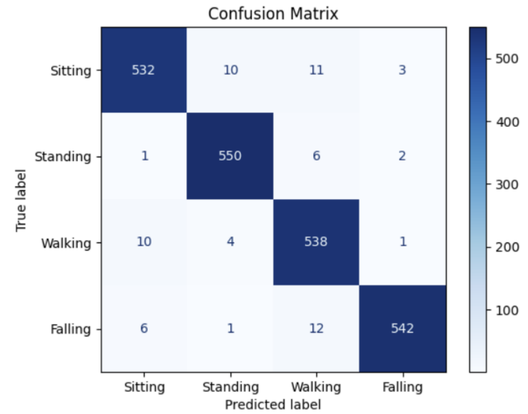
- **Data Preprocessing:**
  - View collected data set



  - Created windowed data frames that
  - hold 15 data points
  - 15 data points allowed us to minimize inference time (0.04s) while maximizing accuracy (97%)
  - Used one hot encoding to encode the labels.
- **Model Optimization:**
  - Build Convolutional Neural Network
  - Add 1D convolutional layer with 64 filters, and kernel size of 3.
  - Reduced the features maps with maxpooling.
  - Used two hidden layers
    - Layer 1 had 100 neurons
    - Layer 2 had 50 neurons
  - Applied a 50% dropout to each layer to prevent overfitting the model
  - Used an output layer with 4 neurons for each classification and a softmax activation function
  - Did not use distillation or pruning because model size and accuracy was sufficient for our embedded system
- **Model Compression:**
  - Converted our model to a tf.lite file [1]
  - Used the default tf.lite optimizations which quantized our weights and activation functions into 8-bit values [1]

Results

- **Accuracy:**
  - Before Quantization: 97.4%
  - After Quantization: 96.9%



- **Model Size:**
  - Before Quantization: 200 KB
  - After Quantization: 50KB
- **Deployment:**
  - 0.04 second inference time
  - Allows our model to make quick real time decisions
- **Demonstration:**
  - Green: Walking
  - Red: Falling
  - Blue: Sitting
  - Yellow: Standing

Discussion and Future Work

- **Discussion:**
  - Accurately detected motion activities with a low inference time
  - Gained experience working with embedded systems and ML
- **Future Work:**
  - Experiment different model types such as LSTM
  - Train on a larger and more accurate dataset
  - Experiment with knowledge distillation and pruning, to possibly reduce size and increase speed

References

- [1] TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from https://www.tensorflow.org/
- [2] Google Colaboratory. (n.d.). Retrieved from https://colab.research.google.com/

Acknowledgements

- Thank you, Dr. Mary Loveless for teaching and guiding us through model development and implementation.
- Thank you to the E222 class for collecting our data set.