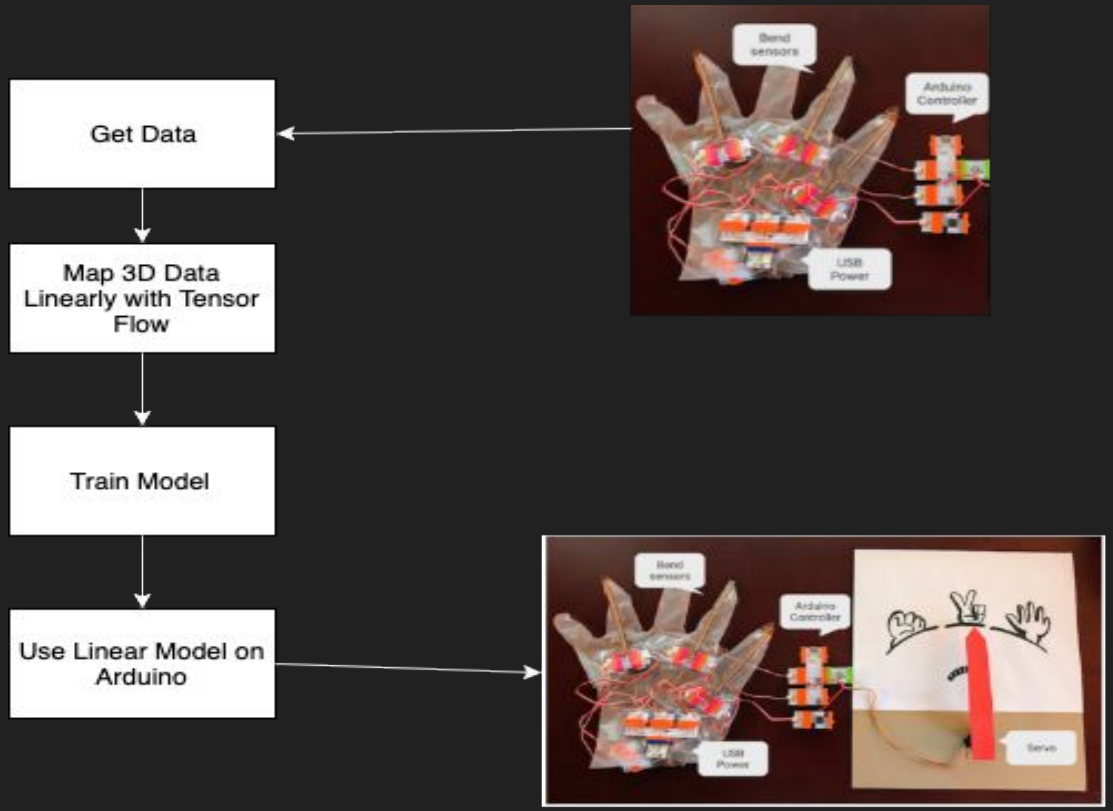
Rock, Paper, Scissors
Machine Learning Application

CMPE 255 Data Mining
Professor Sithu Aung
Krishna Priya Gajula
Brian Hoang
Anjani Mallampati
Archana Yadawa

Introduction

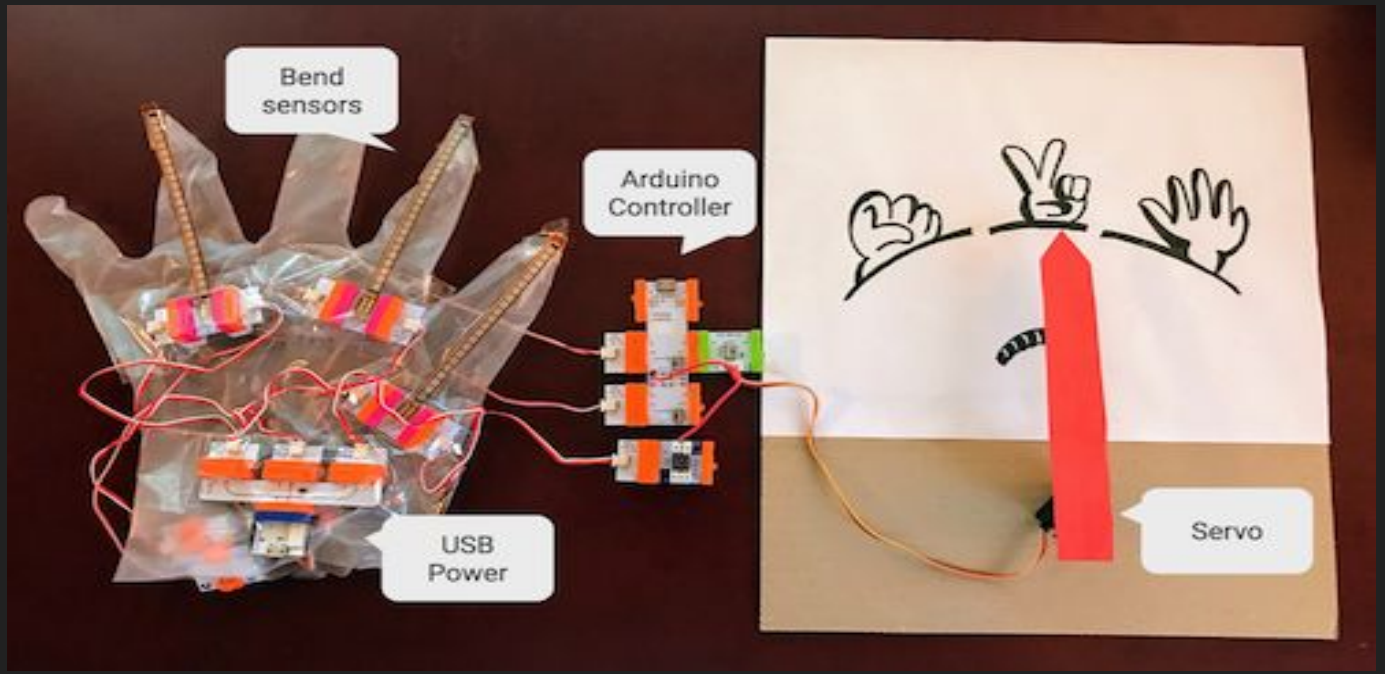
Rock, Paper, Scissors application utilizes both hardware and software components to capture/predict responses for a typical game

High Level Diagram



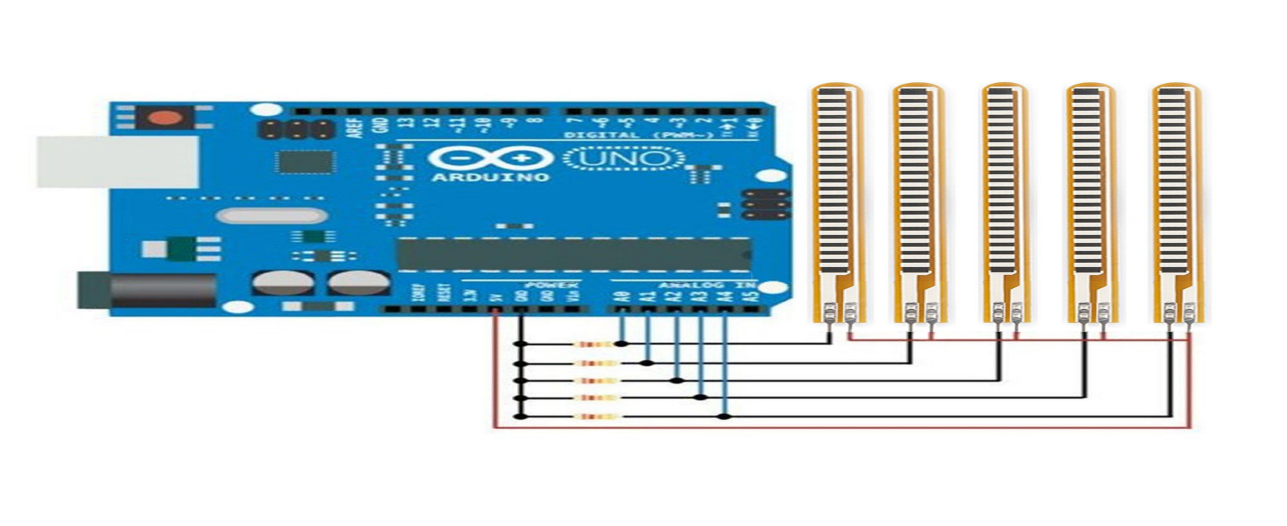
Hardware Setup 1

Overall hardware



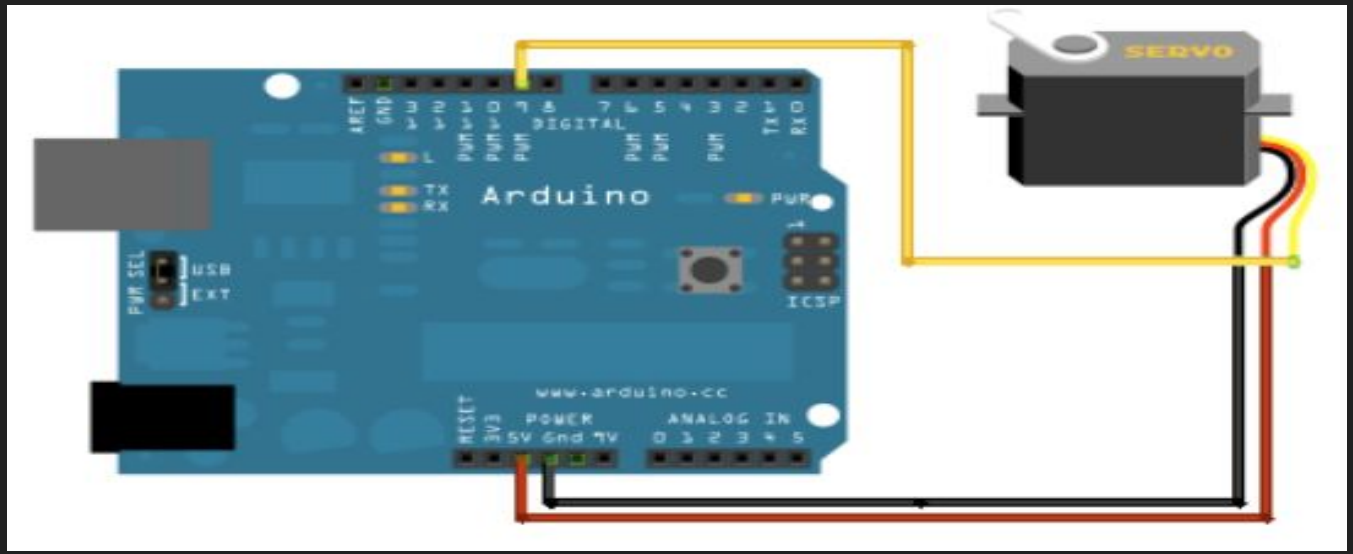
Hardware Setup 2

Connection between flex sensors and Arduino microcontroller:



Hardware setup 2:

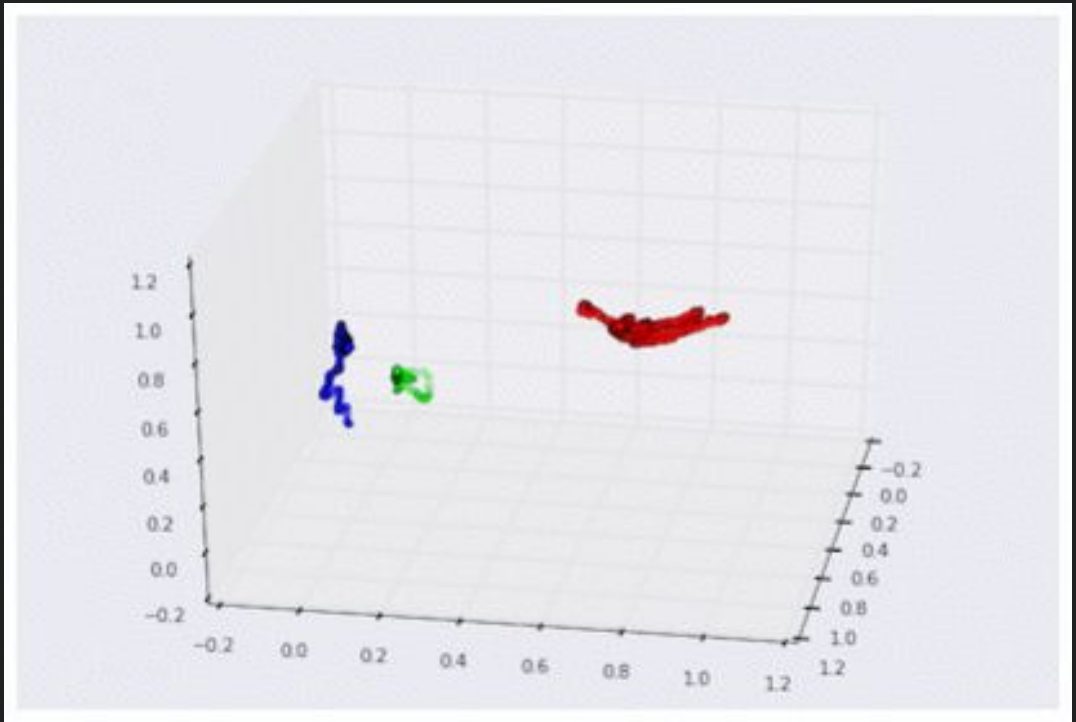
Connections between a Arduino microcontroller and Servo motor:



Step 1 : Getting the Data

- Coding is done on the Arduino module so that it reads data from the bend sensors
- The data is read from the flex sensors every 0.1s and then logged on the serial console
- The Arduino module converts the input signal voltage (0V - 5V) to numbers ranging from 0 to 1023
- The data of rock, paper, and scissors is stored in three different csv files

Step 2 : Visualizing the Data



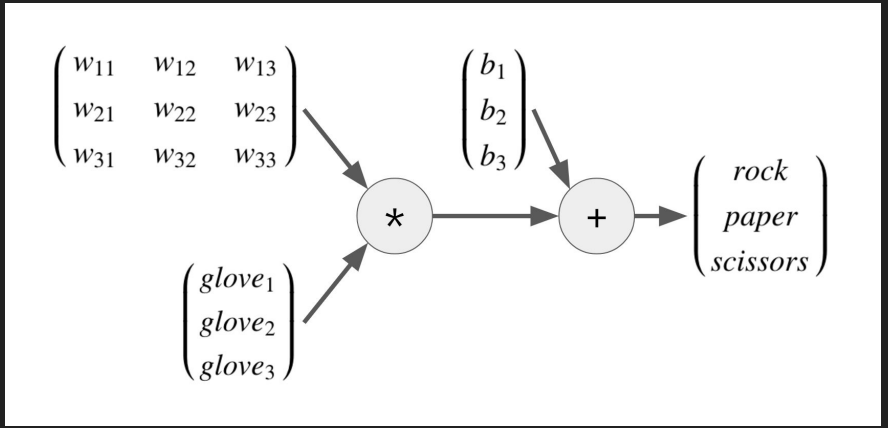
Step 3: Linearly Mapping 3D Data with TensorFlow

- Significant to this project as it allows dynamic determination of dataset values (without need to hardcode ranges).
- TensorFlow attempts to find the best weights and biases that will fit into the linear transformation by calculating them backward in the graph (training the model).
- Formula when transforming 3D data into a linear representation:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Step 3: Linearly Mapping 3D Data (cont.)

- y = wx + B
- 3 x 3 weight matrix dynamically determined by Tensorflow algorithms.
- 1 x 3 glove matrix representation of data gathered by sensors. Represents the linear equations' independent variable.
- Bias dynamically determined by predicting any potential skewing present in dataset.



Step 4 : Training the Model

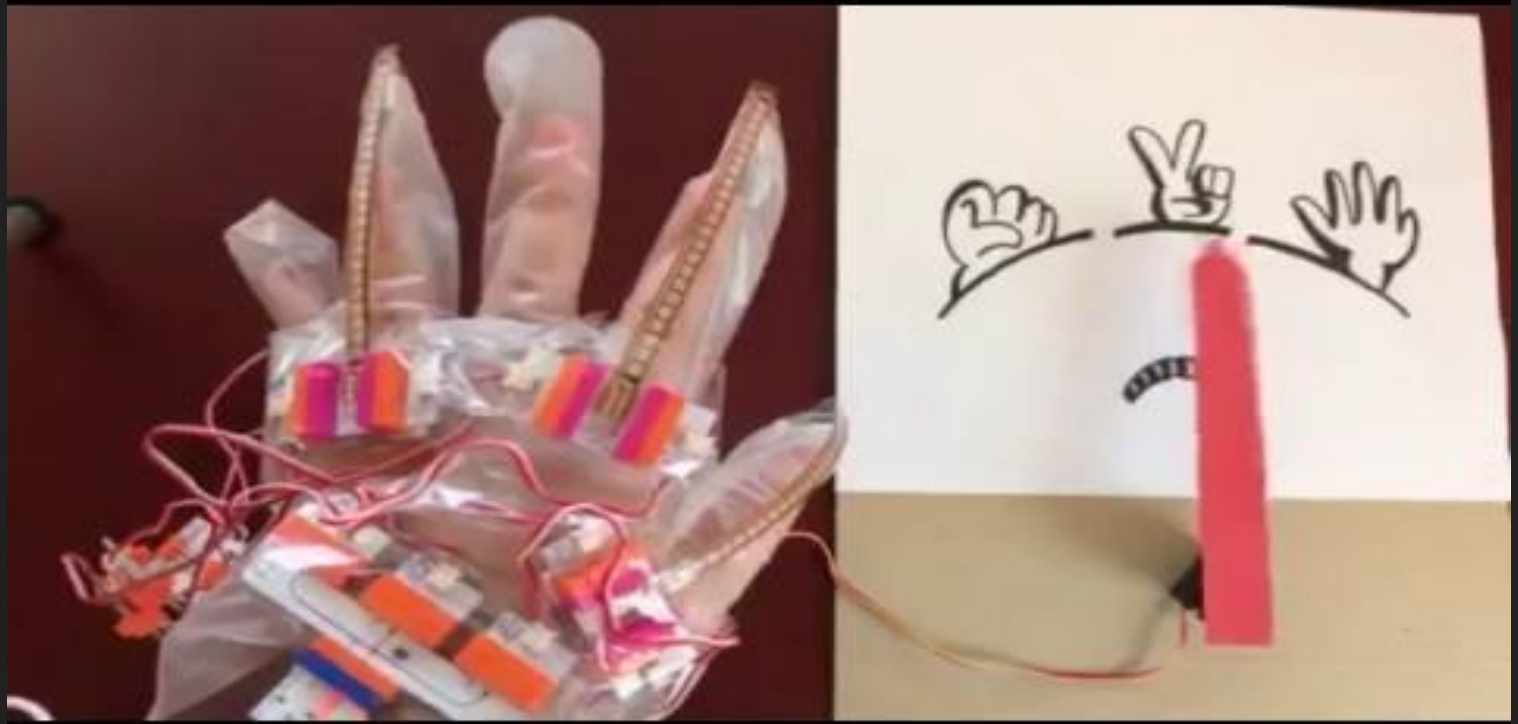
- Define a “coach” for the training
  - Pass hundreds of labels (one-hot labels) for each glove sensor data
    - [1 0 0] for rock
    - [0 1 0] for paper
    - [0 0 1] for scissors
  - Define the loss function (how much error the model has) with a combination of softmax and cross entropy
    - Softmax: squish data into range of 0.0 to 1.0 -- estimated probabilities
    - Cross entropy: returns difference between one-hot labels and estimated probabilities
  - This function will create the “coach” for Tensorflow that will guide it to find the best parameters

Step 4 : Training the Model (cont.)

- Train the Linear Model
  - Run Tensorflow to train the model using an optimizer - Gradient Descent algorithm
  - To run the actual training:
    - create a Session
    - call run method to pass the glove sensor data and labels to the optimizer
    - call run method thousands of times
  - Result of training: good set of weights and biases that can map the glove sensor data into the rock paper scissors space with softmax probabilities

Step 5 : Use Linear Model on Arduino

- Weighted data from the linear model is fed into the Arduino
- Then the linear model on the Arduino maps the glove sensor data into the rock paper and scissors space by doing matrix multiplication on the weights and biases
- From the matrix multiplication we can get the highest value to predict whether it is rock paper or scissors
- Depending on the prediction of the input, the Arduino will determine which output the servo should point to in order to win



References

https://github.com/kazunori279/ml-misc/tree/master/glove-sensor

https://cloud.google.com/blog/products/gcp/my-summer-project-a-rock-paper-scissors-machine-built-on-tensorflow

Thank You