Progress Report P&G Servos

Cody Kesler, Parker Hamilton, Ryan Day, Spencer Yu

November 26, 2019

Our original intent for the project was to build an LSTM Deep Learning model that could take time series data and predict when a failure of the line was going to happen. Given time constraints and issues with securing the data we have adapted the goals and scope of the project. After discussion with Steve and Andy from Proctor and Gamble, Dr. Warnick, and between ourselves, we have decided to concentrate the effort of the project to the two following objectives:

- Compare line effectiveness based on features such as total up-time and number of ramp up and ramp down times for each line. As part of this we will attempt to define a measure for whether a line is effective or not.
- Predict whether a run's up-time will be greater than a cutoff "success" value, based on servo data for the first 10 minutes of a run.

Objective 1: To complete the first objective, we are analyzing the discrete probability density for the length of up-times of each run. These seem to follow an exponential distribution overall. That indicates that there are many failed runs to one successful long steady state run. We are going to use the distribution parameter of an exponential MLE fit to each line as a score, which will be informative if every line fits an exponential distribution well. This is more informative than using other statistics, such as a mean and variance of run times, as the distributions are not normal, and the single exponential parameter tells us how much probability mass is located at longer run times between distributions.

Objective 2: To accomplish our second objective, we have developed a neural network for prediction. We chose 1.5 hours as the cutoff for a "good" run. Each data point is a ramp up time for which the up-time was longer than 10 min. The output label is True/False for longer or shorter than 1.5 hours. The data includes temperature, torque, velocity error, and position error measurements for each machine. There are eleven machines in each line, and there are 10 points in time where measurements are taken. With all of these factors there are 440 features per data point describing the first 10 minutes of a run. Additional features are added to identify which line the data point is from. With the data of six lines there are roughly 2400 runs with 447 features each to train the network on.

Neural Net Details: We built an initial model, and by the project due date we will tune parameters to achieve better accuracy. We implemented the neural net using tensorflow. In the current build, we use four layers with node counts of 447, 1341, 1341, and 1. The first layer is completely dense and the second has a dropout rate of .2. The activation function for internal layers is ReLU, with a final softmax activation function. The data is partitioned into a 80% training, 20% test split, with the servo features distributed evenly in both sets. We initially set the learning rate to a relatively high .05. Our initial model has a classification accuracy of about .62, which is unsurprising considering the sparseness of the data. Ideally, we would have second-frequency time measurements of servo metric at all times, not just when the servo is ramping up or down, but we are working with the data we have. We will further tune hyperparameters and net structure to attempt to improve performance by the project due date.