AI for Experimental Controls

Diana McSpadden dianam@jlab.org,
Torri Jeske roark@jlab.org
Naomi Jarvis, Thomas Britton, Kishansingh Rajput, and David Lawrence
AI Town Hall, July 26th 2021
AIEC Goals:

- **AI-recommended HV settings to maintain GlueX central drift chamber gain**
  - Chamber gain is sensitive to atmospheric pressure

- **Have neural network determine calibration constants as quickly as possible**
  - Reduce time for offline calibration

- **Application to other detector systems such as CLAS12 spectrometer**
Can we use a neural network to predict existing gain constants to within 1%?
The AI Process:

Input Features:

- Aggregate features per run from experimental data and EPICS system:
  - Netamp = pulse height - pedestal, momentum, track angle, drift time

- Split data into train and test sets:
  - 438 runs from 2018
    - 350 train
    - 88 test
  - 897 runs from 2020
    - 717 train
    - 180 test

Feature Importance:

Example output of average importance of features using the `shap python library`. 
Current status:

- 4 Tensorflow sequential neural networks have been hyperparameter tuned based on each set of features
  - Contain varying number of input features
  - Contain 2 or 3 layers
  - Use sigmoid and/or relu activation functions
- Success metric: within 1% max absolute error percentage

All 2020 runs perform within 1%

97.8% of 2018 test data perform within 1%
What’s next:

- Near term:
  - Predicting time to distance calibration constants
  - Improving gain model(s)
  - Verifying predicted gain constants

- Intermediate term:
  - Collecting data at slightly different HV setting

- Long term:
  - Deployment for GlueX
  - Application to CLAS12 drift chambers

Please send all questions/comments/suggestions via email :)

6
backups
Training and Test Data Set Information

1. runs_epics_gain2018.csv == all 2018 data (count: 438)
2. runs_epics_gain2020.csv == all 2020 data (count: 897)
3. test-holdout2018.csv == final testing/holdout dataset, only 2018 data (count: 88)
4. test-holdout2020.csv == final testing/holdout dataset, only 2020 data (count: 180)
5. test-holdout20182020.csv == union of test-holdout2018.csv and test-holdout2020.csv (count: 268)
6. train-validate2018.csv == training dataset for training and validation, only 2018 data (count: 350)
7. train-validate2018w.csv** == training dataset for training and validation, only 2018 data (count: 412)
8. train-validate2020.csv == training dataset for training and validation, only 2020 data (count: 717)

** weighted, i.e. upsampled 2018 training data
Models Discussed

- "11 Feature Vanilla LR": Linear Regression model with following features:
  - 'Pressure',
  - 'netamp.CV', 'netamp_mean', 'netamp_75%', 'netamp.std', 'netamp_25%', 'netamp_50%'
  - 'theta.std', 'theta.75%'
  - 'p.25%', 'p.50%'

- "11 Feature NN": Sequential Neural Net model:
  - 'Pressure'
  - 'netamp.CV', 'netamp_mean', 'netamp_75%', 'netamp.std', 'netamp_25%', 'netamp_50%'
  - 'theta.std', 'theta.75%'
  - 'p.25%', 'p.50%'

- "No Mode NN": Sequential Neural Net model with no mode features:
  - 'Pressure', 'Current',
  - 'eventCount',
  - 'netamp.std', 'netamp_25%', 'netamp_mean', 'netamp_75%', 'netamp_50%', 'netamp.max', 'netamp.CV'
  - 'theta.std', 'theta.min', 'theta.mean', 'theta.25%', 'theta.50%', 'theta.CV'
  - 't.mean', 't.std', 't.min', 't.25%', 't.50%', 't.max', 't.CV'
  - calculated features:
    - 'netamp.additive' = 'netamp.25%' + 'netamp.50%' + 'netamp.75%' + 'netamp.max'
    - 't.additive' = 't.min' + 't.25%' + 't.50%' + 't.max'
    - 't.range' = 't.max' - 't.min'
    - 'p.range' = 'p.max' - 'p.min'

- "37 Feature NN": Sequential Neural Net model - results currently only available on 2018 data:
  - 'Pressure', 'Current',
  - 'event_count'
  - 'netamp.std', 'netamp.CV', 'netamp_mean', 'netamp.75%', 'netamp_50%', 'netamp_25%', 'netamp_max'
  - 'theta.CV', 'theta.75%', 'theta.25%', 'theta.50%', 'theta.mean', 'theta.mode', 'theta.min', 'theta.std'
  - 't.CV', 't.50%', 't.75%', 't.25%', 't.additive', 't.min', 't.max', 't.range', 't.std', 't.mean'

- "39 Feature NN": Sequential Neural Net model - results currently only available on 2018 data:
  - 'Pressure', 'Current',
  - 'eventCount'
  - 'netamp.std', 'netamp.CV', 'netamp_mean', 'netamp.75%', 'netamp_50%', 'netamp.additive', 'netamp.25%', 'netamp.max', 'netamp.mode',
  - 'theta.CV', 'theta.75%', 'theta.25%', 'theta.50%', 'theta.mean', 'theta.mode', 'theta.min', 'theta.std',
  - 't.CV', 't.50%', 't.75%', 't.additive', 't.min', 't.max', 't.range', 't.std', 't.mean"
Understanding importance of an individual feature in a multivariable, non-linear function?

For regression problems, there are a number of available evaluation methods.

We implemented Shapley values.

“The Shapley value is a framework originally proposed in the context of game theory to determine individual contributions of a set of cooperating players” - Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications | IEEE Journals & Magazine | IEEE Xplore