Development of methods for Continuous Learning in the realm of Dynamic Predictive Maintenance

Vrishank Bhardwaj
Nikhil Panse

Data Description

FD0001 -

- Sea Level Operating Conditions
- HPC Degradation - Single mode of failure
- Training data for 100 engines operating till failure
- Testing data for 100 engines operating till failure

FD0003 -

- Sea Level Operating Conditions
- HPC and Fan Degradation - Two modes of failure
- Training data for 100 engines operating till failure
- Testing data for 100 engines operating till failure

Goal -

Develop and test approaches for Continuous Learning for Predictive Maintenance on CMAPPS dataset

Metric -

RMSE for RUL predictions

Methodology

- Split FD0001 training data in 4 subsets of 25 engines each
- Split FD0003 training data in 4 subsets of 25 engines each
- Total of 8 batches available for Continuous Learning Experiment training data
- Training batches are made available to model successively
- After each batch, model is tested on testing data

Methodology

- First 4 batched fed belong to FD0001 (Single mode of failure)
- Only FD0001 testing set is used for the first four evaluations
- Now, the next four batches from FD0003 (Two modes of failure) are successively made available to the model
- Here, Performance on Testing data from FD0001, FD0003 and the combined testing data is measured separately
- Since order of batched during training may affect the results, the RMSE is averaged over 10 runs with randomised orders

Baselines to be established

- Naive approach - Only current batch is available for training
- Cumulative approach - All past batches can be used for training

Our Approach

1. Dynamic Weighting across models
    a. New model is trained on each new batch of data
    b. All such models are stored
    c. Lightweight linear model is trained after each episode to find optimal weights for predictions of these models
    d. Predictions are combined using weighted average (using weights from step-c)
2. Dynamic Meta-Model
    a. New model is trained on each new batch of data
    b. All such models are stored
    c. Tree based model(Random Forest Regressor/Gradient Boosted Regressor) is trained on new data combined with predictions of previous models on new data, i.e. predictions of previous models used to train new model
    d. Final output from Tree based model is used for evaluation

Excerpt from Core50 supporting Baseline

When addressing real-world continuous learning, assuming that the whole past data can be used at each training step (i.e., cumulative approach) is not only very far from biological learning but also unlikely for application engineering. In fact, the cumulative approach would require:

- To store all the previous data streams
- To retrain a model on all the whole data each time new data is available

Updating an already trained model with new data (only) is much more feasible in term of computation and memory constraints. Recent advances in transfer learning/tuning with deep neural networks have shown that using previously learned knowledge on similar tasks can be useful for solving new ones [15]. Yet, little has been done in the context of continuous learning where the same model is required to solve new tasks while maintaining good performances on the previous ones. Indeed, preserving previously learned knowledge without re-accessing old patterns remains particularly challenging due to the phenomenon known in literature as catastrophic forgetting where what has been learned so far is seriously compromised.