Predictions from Learning Curves, Hyper-parameters and meta-features

Barke Saiprasad: 4559259

Project goals

- Experiment with different types of data scaling and transformations
- Experiment with different architectures and compare their performances
- Use hyper-parameter optimization technique BOHB to tune the network and find optimal λ* for the task.

Datasets

- **Task B**: 2000 configurations applied on 6 data sets, meta features of the 6 datasets and run time information up-to 50 epochs.

Data Used

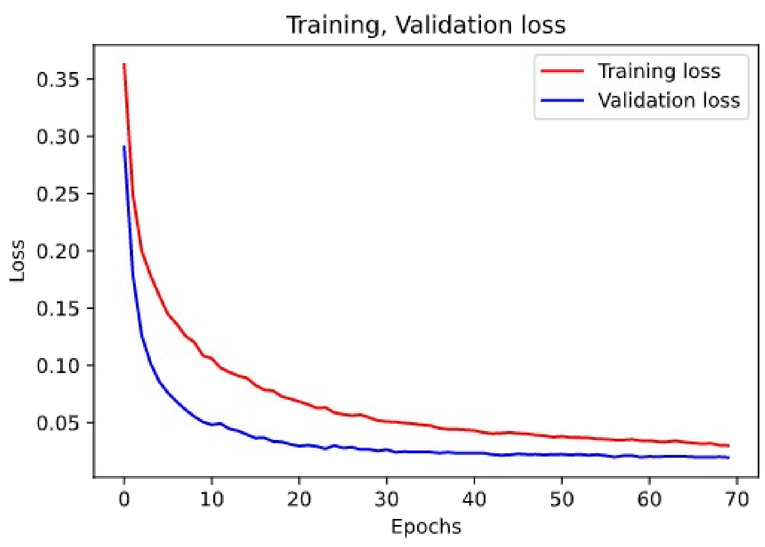
- **Task B**: 7 hyper-parameters for the 2000 configurations applied on 6 datasets and final validation accuracy in each case as the target.
- **Min-Max scaling** was applied to input variables in order to improve the convergence performance

Methodology.

- **Task B: Regression Problem**.
  - **1.** The meta-features of each data set and the configurations are concatenated to make unique feature vectors which are then used to regress the values of the final validation accuracy using **MLPs** with different hyper parameters. Tuning is performed manually without any HPO method
  - **2**. Flexible MLP network with variable batch size tuned using **BOHB**.
  - **3. AutoPytorch**

Experiments and Results

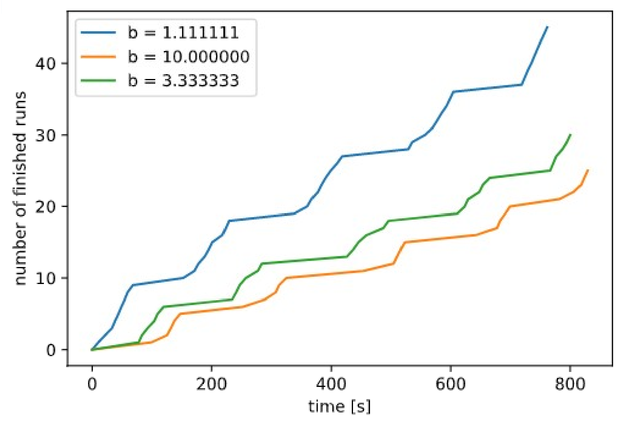
- **Training Validation Curve for simple MLP**

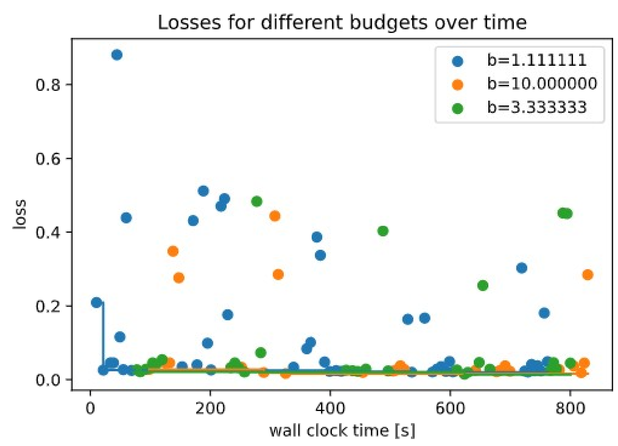


Model:
Sequential(
  (fc1): Linear(in_features=55, out_features=41, bias=True)
  (dropout1): Dropout(p=0.5, inplace=False)
  (relu1): ReLU()
  (fc2): Linear(in_features=41, out_features=27, bias=True)
  (dropout2): Dropout(p=0.5, inplace=False)
  (relu2): ReLU()
  (fc3): Linear(in_features=27, out_features=13, bias=True)
  (dropout3): Dropout(p=0.5, inplace=False)
  (relu3): ReLU()
  (fc4): Linear(in_features=13, out_features=1, bias=True)
)
Final Test Loss: 0.0687

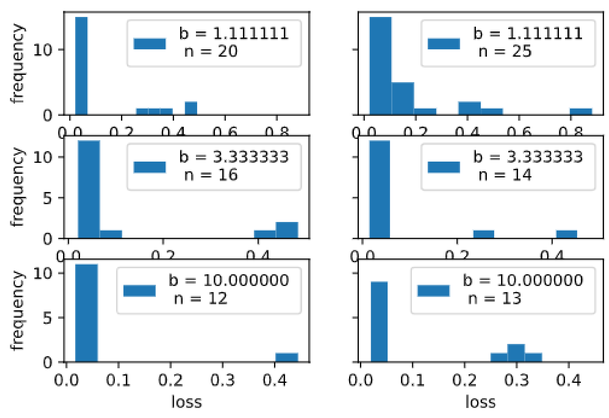
BOHB statistics

-





Loss of model based configurations (left) vs. random configuration (right)



A total of 60 unique configurations where sampled.
A total of 80 runs where executed.

The best model (config_id (5, 0, 2)) has the lowest final error with 0.0168 and **test error 0.0301.**
Sequential(
  (fc1): Linear(in_features=55, out_features=41, bias=True)
  (dropout1): Dropout(p=0.2237009733229576, inplace=False)
  (relu1): ReLU()
  (fc2): Linear(in_features=41, out_features=27, bias=True)
  (dropout2): Dropout(p=0.2237009733229576, inplace=False)
  (relu2): ReLU()
  (fc3): Linear(in_features=27, out_features=13, bias=True)
  (dropout3): Dropout(p=0.2237009733229576, inplace=False)
  (relu3): ReLU()
  (fc4): Linear(in_features=13, out_features=1, bias=True)
)
{'batch_size': 2, 'dropout_rate': 0.2237009733229576, 'lr': 0.0013084672707647368, 'num_layers': 4, 'optimizer': 'Adam'}

Auto-Pytorch

- {'optimized_hyperparameter_config': {'CreateDataLoader:**batch_size':** **125**, 'Imputation:strategy': 'median', 'InitializationSelector:initialization_method': 'default', 'InitializationSelector:initializer:initialize_bias': 'No', **'LearningrateSchedulerSelector:lr_scheduler':** **'cosine_annealing'**, 'LossModuleSelector:loss_module': 'l1_loss', 'NetworkSelector:**network':** **'shapedresnet',** 'NormalizationStrategySelector:**normalization_strategy':** **'standardize',** 'OptimizerSelector:optimizer': 'sgd', 'PreprocessorSelector:preprocessor': 'truncated_svd', 'ResamplingStrategySelector:over_sampling_method': 'none', 'ResamplingStrategySelector:target_size_strategy': 'none', 'ResamplingStrategySelector:under_sampling_method': 'none', 'TrainNode:batch_loss_computation_technique': 'standard', **'LearningrateSchedulerSelector:cosine_annealing**:T_max': **10**, 'LearningrateSchedulerSelector:cosine_annealing:eta_min': **2,** 'NetworkSelector:shapedresnet:**activation':** **'relu',** 'NetworkSelector:shapedresnet:**blocks_per_group':** **3,** 'NetworkSelector:shapedresnet:**max_units': 118,** 'NetworkSelector:shapedresnet:num_groups': 3, 'NetworkSelector:shapedresnet:resnet_shape': 'brick', 'NetworkSelector:shapedresnet:use_dropout': 0, 'NetworkSelector:shapedresnet:use_shake_drop': 0, 'NetworkSelector:shapedresnet:use_shake_shake': 0, 'OptimizerSelector:**sgd**:**learning_rate':** **0.017296045778574753,** 'OptimizerSelector:**sgd:momentum':** **0.7179993052839114,** 'OptimizerSelector:**sgd:weight_decay':** **0.010928546818963084**, 'PreprocessorSelector:truncated_svd:target_dim': 100}, 'budget': 1.0, 'loss': 0.06484225392341614, 'info': {'loss': 0.10131112998351455, 'model_parameters': 263849.0, 'train_mean_abs_error': 0.10131112486124039, 'lr_scheduler_converged': 0.0, 'lr': 0.017296045778574753, 'val_mean_abs_error': 0.06484225392341614}}

Final loss with autopytorch is 0.0731.