Practical Machine Learning

Wildcard Project

(Electrical Load Forecasting using Deep Neural Network)

Sanjeeb Humagain

May 4, 2024

Introduction:

Load forecasting is an important task in power system because of its impact on energy management system. There are three types of load forecasting, short term, medium term and long term load forecasting. Short term load forecasting (STLF) covers the load forecasting from hour to a few weeks. Accuracy of load forecast is crucial to reduce the loss due to error in load forecasting. During the last couple of decades, artificial intelligence (AI) has come out as one of the most distinguished areas of research because of its ability to make systems automatically take actions to increase efficiency as well as performance [1]. AI makes it possible for a system to create a model and train it with set of instructions or data to learn and then use the model for test data and future data to make decisions. The use of AI is increasing rapidly in every sector from daily life of a person to the large industries. With the advancement of technology, the security risk also increases and it is always important to make a system robust in order to implement in real life.

Therefore, there has been a significant amount of research conducted on load forecasting using artificial intelligence (AI) methods.

Deep learning (DL) is a kind of machine learning (ML) algorithm which consists of multiple hidden layers enabling the ML model to learn fast and handle the non-linear relationships with parameters that affects the load demand. DL is used in STLF in order to increase the accuracy of forecast.

Additionally, the accuracy of load forecasting has a significant impact on costs. There are several advantages to accurate load forecasting, including the reduction in operating and maintenance costs, enhancement of power system reliability, and improvement in planning and decision making for electrical load dispatch. Overestimation of load causes an increase in the reserve capacity, which is not required, thereby increasing operating costs. Similarly, underestimation of load causes in a failure to supply the required spinning and standby reserve, potentially leading to stability issues on voltage and frequency, and even collapse of the power system in extreme cases [2].

In this exercise, we will try to implement deep learning and see whether the performance increase or not. We will compare the result of deep neural network with 3 hidden layers with the result of NN with single hidden layer.

We will compare the performance of deep neural network using random search and Bayesian hyperparameter optimization techniques and suggest an optimization based on the result.

Following libraries are used for this task:

- pandas
- seaborn
- matplotlib.pyplot
- numpy
- train_test_split
- BayesianOptimization from bayes_opt
- mean_absolute_error, mean_squared_error, r2_score from sklearn.metrics
- RandomForestRegressor
- RandomizedSearchCV

Some of the libraries used are listed above. They are used for data preprocessing, numerical computation, forecasting, optimization and performance evaluation of the model.
In all ML algorithms used, roughly same steps are followed. They are, checking for the missing value, data preprocessing, clean up, scaling, split up, creating model, training the model and running the test, and hyperparameter optimization etc. After performing the hyperparameter optimization, the same model was run with the optimized set of values of the hyperparameters. Finally, the performance of the model and hyperparameters are summarized and discussed by comparing mean square error, mean absolute error and r2 score of each of the options considered in this experiment.

Dataset Description:

In this experiment, the dataset has 9 features which are used to predict the system load. The data set contains 87648 samples of data and the target system load from 2006 to 2011. In this exercise, the system load data is being used to compare machine learning algorithms and optimization methods.

Experimental Setup:

Python is used for this experiment because of the availability of crucial libraries for instance Pandas (for data manipulation), Scikit-learn (for ML) etc. The dataset imported from the csv file and split into training and testing sets in ratio of 80:20 in the same way as performed in previous exercises.

Deep Neural Network (DNN) is used for this exercise which has three hidden layers. Similarly, Bayesian optimization method is implemented for hyperparameter optimization process. The performance of model model was assessed using mean square error, mean absolute error and r2 score.

Nested resampling is a process that provides reliable estimate of a model's performance and assists to prevent overfitting during the entire process of hyperparameter tuning. Specially, it is important when evaluating the performance of various models or even when comparing the different techniques for hyperparameter optimization. In the code, first the training and testing data are split in ratio of 0.8/0.2 which works as outer loop for model evaluation. Hyperparameter tuning is performed on the training folds, while the validation fold is used to evaluate the performance of

the model with the selected hyperparameters. Total 100 number of iterations were performed for Bayesian Optimization.

Additionally, inner resampling is performed during the process of training within the train_model function. Specifically, inside the for loops, (for inputs, labels in train_loader:) and (for inputs, labels in test_loader:) for each every epoch. The model is trained on the training dataset ('train_laoder'), and the performance is evaluated on the validation dataset ('test_loader').

Results:

The MSE, MAE and R2 score of after using Bayesian optimization methods are listed below.

Mean Squared Error (MSE): 200479.8747

Mean Absolute Error (MAE): 348.0457

R2 Score: 0.9003

Talking about the range of hyperparameters used, the values are tabulated in table1.

S.N.	Hyperparameters	Deep Neural Network ML algorithm	
		Bayesian Optimization	
		Range defined	Tuned value
1	hidden_size1	(8, 256)	64.81
2	hidden_size2	(8, 128)	120.44
3	hidden_size3	(8, 128)	94.66
4	num_epochs	(10,100)	98.30
5	lr	(1e-5, 1e-1)	0.0092

Table 1: Hyperparameter values after Bayesian HPO

Considering the time of execution in previous exercises, Random search is doing better job for optimization in both RF and NN. However, my research is to use deep neural network for load forecasting to improve the system reliability that is why I am using deep neural network in this exercise.

The Plots generated are listed below.

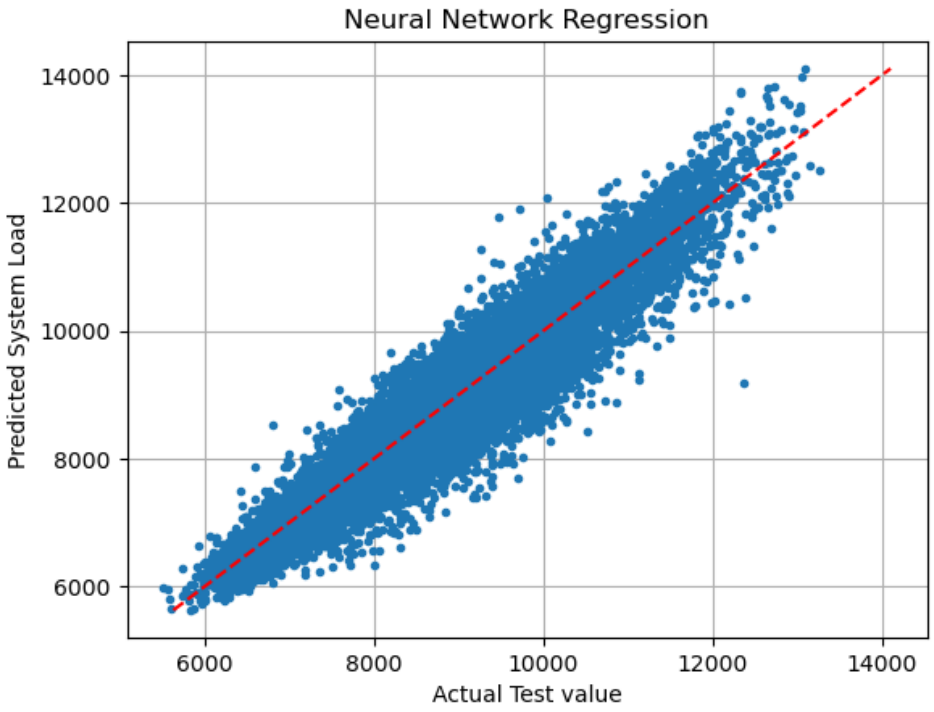

Fig1: Actual vs predicted value using Bayes HPO for DNN

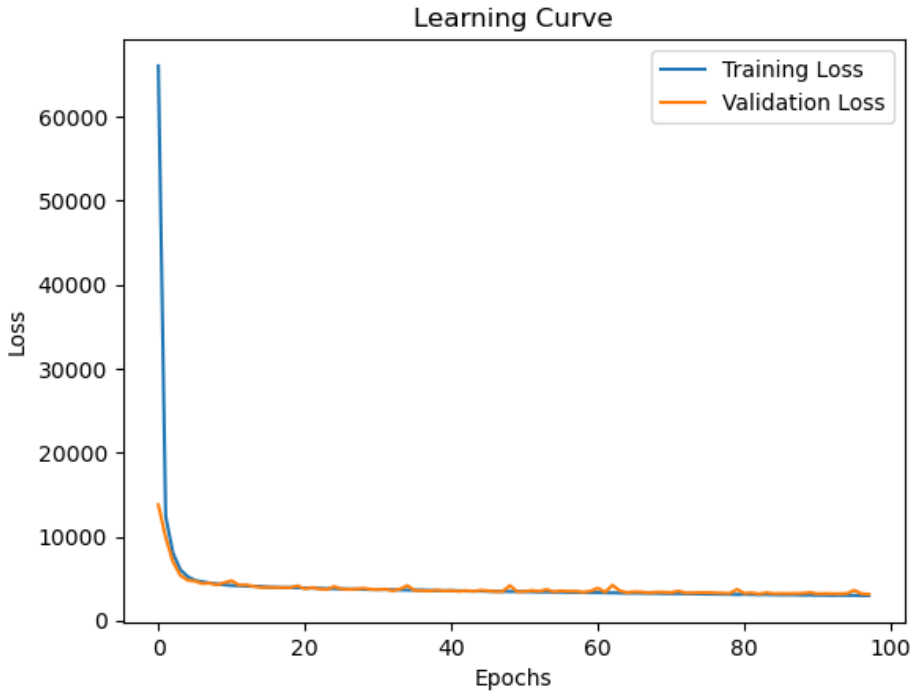

Fig2: Learning Curve for DNN model

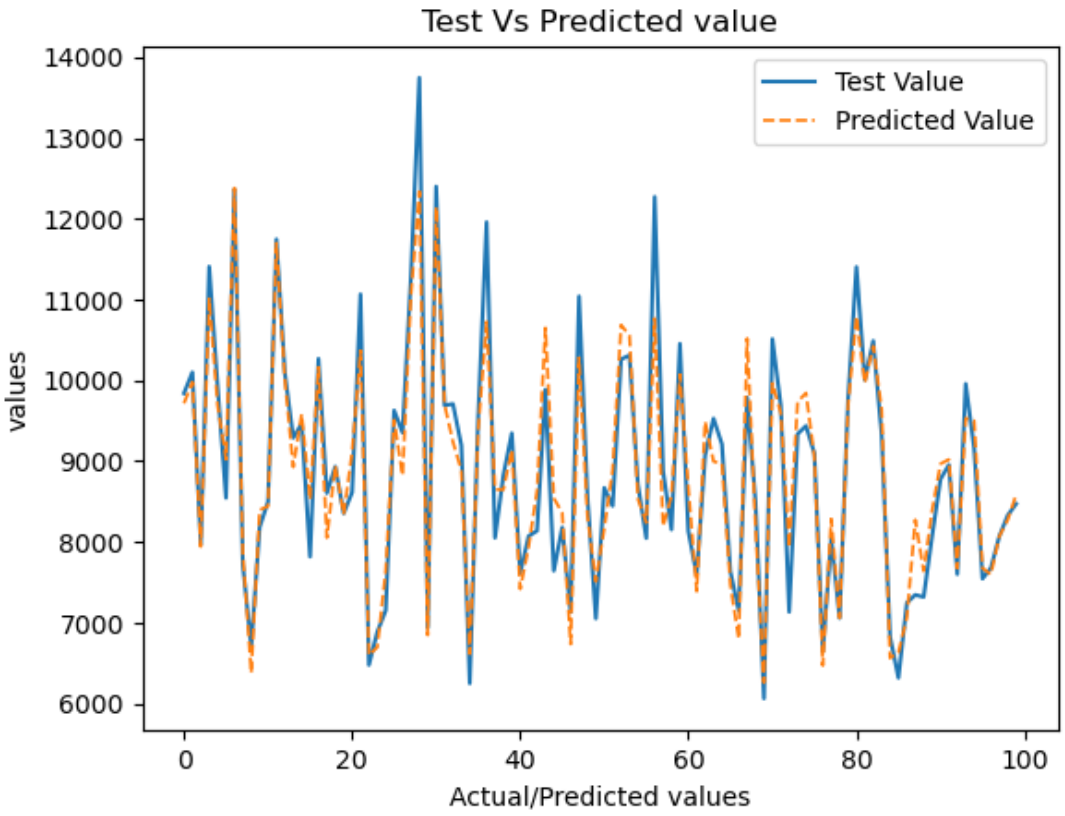

Fig3: predicted and actual data plot for DNN model

Code: Code is available on the github repository on the following link.

https://github.com/COSC5557/wildcard-project-Sanjeeb-PL.git

References:

1. S. Zhao, F. Blaabjerg and H. Wang, "An Overview of Artificial Intelligence Applications for Power Electronics," in IEEE Transactions on Power Electronics, vol. 36, no. 4, pp. 4633-4658, April 2021, doi: 10.1109/TPEL.2020.3024914.
2. Singh, Saurabh, Shoeb Hussain, and Mohammad Abid Bazaz. "Short term load forecasting using artificial neural network." 2017 Fourth International Conference on Image Information Processing (ICIIP). IEEE, 2017.