PROJECT REPORT: Hyperparameter Optimization using Bayesian Optimization with Gaussian Process Regression

1. INTRODUCTION

Hyperparameter optimization is a critical task in machine learning where the objective is to find the best of hyperparameters for a given model to achieve optimal performance on unseen data. Hyperparameters are parameters that are not directly learned by the model during training but affect the learning process (e.g., learning rate, number of estimators in an random forest).

In this project we explore popular methods for hyperparameter optimization: Random Search, Using the hyper-opt library, Bayesian Optimization with Gaussian Process Regression (GPR). The goal is to compare these methods in terms of their effectiveness and efficiency in optimizing hyperparameters for machine learning.

In the following project, I have used the Breast Cancer Wisconsin (Diagnostic) dataset which is available online. The dataset consists of features computed from digitized images of breast mass and aims to predict whether a tumor is benign or malignant based on these features.

2. METHODOLOGY

Data Preprocessing

- Data Loading: The Breast Cancer dataset is loaded using the 'sklearn.datasets.load_breast_cancer()'.
- Data Cleaning: Missing values, if any, are handled by dropping rows with missing data by using 'dropna'.
- Feature Encoding: Categorical variables (if any) are encoded using 'LabelEncoder' to convert them into numerical format.
- Feature Scaling: Features are scaled using 'StandardScaler' to standardize them around a mean of zero and a standard deviation of one.
- Train-Test Split: The dataset is split into training (80%) and testing (20%) sets using 'train_test_split' from 'sklearn.model_selection'.

Model Selection

- Three machine learning models are selected for evaluation based on their suitability for binary classification tasks and their popularity in practical applications.
- Random Forest Classifier: Ensemble learning method that constructs multiple decision tress and merges them together to improve performance.
- Support Vector Classifier (SVC): Effective for high – dimensional data, constructs a hyperplane or set of hyperplanes in a high-dimensional space to classify data points.
- Logistic Regression: Linear model for binary classification that uses a logistic function to model the probability to certain class.
- But In the notebook I have just performed the code on random forest classifier , if you want to run the other models you can change the 'random forest' to other models name in the evaluate and training section.

Hyperparameter Optimization

- Random search: Hyperparameters are sampled randomly from predefined ranges. Each set of hyperparameters is evaluated using cross validation to estimate the model's performance. This process is repeated for a fixed number of iterations or until convergence criteria are met.
- Bayesian Optimization with Gaussian Process Regression(GPR): Bayesian Optimization sequentially selects hyperparameters based on their potential to improve the objective function (e.g., ROC AUC score). It models the objective function using a Gaussian Process which provides a probabilistic estimate of the function's behaviour. An acquisition function guides the search by balancing exploration and exploitation. It iteratively evaluates a new set of hyperparameter, updates the Gaussian Process model, and selects the next set of hyperparameters based on the Acquisition function.
- Hyperopt Library: Hyperopt is employed to perform Bayesian optimization using tree-structured parzen estimator(TPE). It explores function and selecting new hyperparmaters based on expected improvement. It is compared with GPR-based Bayesian optimization and random search to evaluate its effectiveness in optimizing the models.

Evaluation Metrics

- ROC AUC SCORE: Receiver Operating Characteristics Area Under Curve (ROC AUC) is used as the primary evaluation metric. ROC AUC measures the ability of a binary classifier to distinguish between classes and is particularly effective for imbalanced datasets like the breast cancer dataset.

3. PERFORMANCE COMPARSION
    - Random Search vs Bayesian Optimization:
        - Bayesian Optimization with GPR outperform random search in terms of efficiency and effectiveness.
        - It requires fewer iterations to converge and achieves higher ROC AUC scores.
        - Balances exploration and exploitation effectively, making it a preferred choice for optimizing complex machine learning models.

    - Hyperopt vs GPR-based Bayesian Optimization:
        - Hyperopt using TPE demonstrates comparable performance to GPR-based Bayesian Optimization.
        - Both methods achieve competition ROC AUC scores similar convergence rates.

4. CHALLENGES AND OBSERVATIONS

    - Computational Efficiency:
        - Bayesian Optimization methods , including GPR and Hyperopt may require significant computational resources , especially in high – dimensional hyperparameter spaces.

    - Exploration vs Exploitation:
        - GPR and Hyperopt effectively balance exploration (searching new areas of the hyperparameter space) and exploitation (exploiting known good areas), influencing optimization outcomes.

5. GRAPHICAL REPRESENTATION

    - Hyperparameter Convergence Plot: Visualizes how hyperparameters evolve over iterations during Bayesian with GPR and Hyperopt and Random Search.

    - ROC AUC vs Iteration Plot: Shows the improvement in ROC AUC score with each iteration for Random Search, GPR-based Bayesian Optimization , and Hyperopt.

    - Learning rate distributions curves: I tried hard to understand what these curves are for through the sample notebook provided in the resources for this project , but was unable to understand them, so i apologise not to include them , so please consider my submission with the learning rate distributions curves.

6. CONCLUSION

Hyperparameter optimization is crucial for maximizing machine learning model performance. Bayesian Optimization methods , particularly GPR-based, demonstrate superior efficiency and effectiveness compared to Random Search and Hyperopt. It efficiently explores the hyperparameter space and achieve higher ROC AUC scores with fewer iterations, making them preferred approaches for optimizing complex machine learning models.

7. REFERENCES

    - https://ar5iv.labs.arxiv.org/html/1807.02811

    - https://www.mdpi.com/2071-1050/14/19/12777

    - Automated Model Tuning (kaggle.com)

    - AutoML | Hyperparameter Optimization