Multi-Objective Optimisation for Smart Energy Grids Using ResNet

Manda Kausthubh

May 2025

1 Introduction

In this project, we apply Bayesian Optimisation (BO) for multi-objective optimisation in a smart energy grid classification task. The primary goal is to jointly optimise two conflicting objectives:

- **Accuracy**: Classification accuracy of a neural network, ratio of correct classification to the total number of data points.
- **Sparsity**: A proxy for model compactness and energy efficiency, Number of parameters of the model that are zero. This metric is very important for studying explainablity of ML models. Here our choice of ML model is a ResNet with variable learning rates and varying depths.

We utilise a ResNet-based black-box model, and guide our search using BO with weighted-sum scalarization of objectives.

2 Methodology

The optimisation loop follows these steps:

1. Sample initial points using Sobol sequences.
2. Evaluate a black-box function returning accuracy, sparsity, and a weighted objective.
3. Fit a `SaasFullyBayesianSingleTaskGP` model using `botorch`.
4. Optimize the acquisition function `qUpperConfidenceBound`.
5. Repeat to refine the surrogate model and locate optimal trade-offs.

The black-box function is defined as:

```
def black_box_function(x, weights):
    try:
        x_np = x.detach().cpu().numpy().reshape(-1)
        y_pred, _, model = NN_prediction(x_np, test_features, train_loader, val_loader)
        sparsity = compute_model_density(model)
        acc = sklearn.metrics.accuracy_score(y_test, y_pred)
        objectives = torch.tensor([1.0 - acc, sparsity])
        return torch.Tensor([acc, sparsity, objectives @ weights])
    except:
        print("Unexpected eval error", sys.exc_info()[0])
        return torch.tensor([1.0])
```

3 SAASBO: Sparse Axis-Aligned Subspace Bayesian Optimization

SAASBO is a Bayesian Optimization technique using a fully Bayesian GP model with a hierarchical prior that promotes sparsity in high-dimensional problems. The core idea is to assume that only a few dimensions are relevant and exploit this sparsity.

Given observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the GP prior is defined over a covariance kernel:

$$k(\mathbf{x}, \mathbf{x}') = \tau^2 \exp\left(-\sum_{d=1}^{D} \lambda_d (x_d - x'_d)^2\right) \tag{1}$$

Here, λ_d is a precision parameter for the d-th input. SAASBO places a sparsity-inducing prior over $\{\lambda_d\}$:

$$\lambda_d \sim \text{Gamma}(a, b), \quad a \ll 1 \tag{2}$$

This encourages most λ_d to be close to zero, effectively removing the corresponding input dimension from the kernel. The posterior over model hyperparameters is inferred using No-U-Turn Sampling (NUTS), enabling uncertainty-aware optimization.

SAASBO is ideal for our setting where the dimensionality of $\mathbf{x}$ is high but only a few components meaningfully influence the objectives.

4 Results and Observations

The black-box function was evaluated using a ResNet classifier whose weights were determined by the input $\mathbf{x}$. The trade-off between accuracy and sparsity was recorded for several weight vectors. However, the Pareto front visualization (Figure 1) indicated an undesirable degenerate behavior.

Inference

The near-vertical Pareto front suggests that the optimization consistently selects models of nearly identical sparsity. This implies a failure in navigating the sparsity-accuracy trade-off. Potential causes include:

- Overly dominant weight vector component.
- Model convergence to a local mode in the posterior.
- Degenerate acquisition optimization due to NaN gradients.

5 Experimental Setup

The code loops through different weight vectors to scalarize the two objectives. At each step, we optimize the acquisition function using multiple restarts and raw samples. Error handling was incorporated to mitigate 'NaN' gradients and numerical instabilities.

To maintain continuity despite optimizer failures (e.g., 'OptimizationGradientError' due to NaNs), we used `try-except` blocks that catch errors during acquisition optimization and skip to the next iteration.

6 Results

Figure 1 shows the resulting Pareto front obtained by plotting the best accuracy vs sparsity points for each weight configuration.

Observations

The Pareto front appears degenerate, with all points collapsed around a narrow band of sparsity and accuracy. This indicates:

- Either the model is not learning meaningful variation with different hyperparameter inputs.
- Or the architecture is too rigid or not expressive enough to reflect trade-offs.
- Improper scaling or overly aggressive regularization in the surrogate model could also contribute.

7 Conclusion

We implemented a multi-objective BO framework to optimize accuracy and sparsity in a ResNet model. While the pipeline is functional and error-resilient, the optimization currently fails to produce a well-separated Pareto front. This points to deeper issues in the model sensitivity or objective definition that need further investigation.

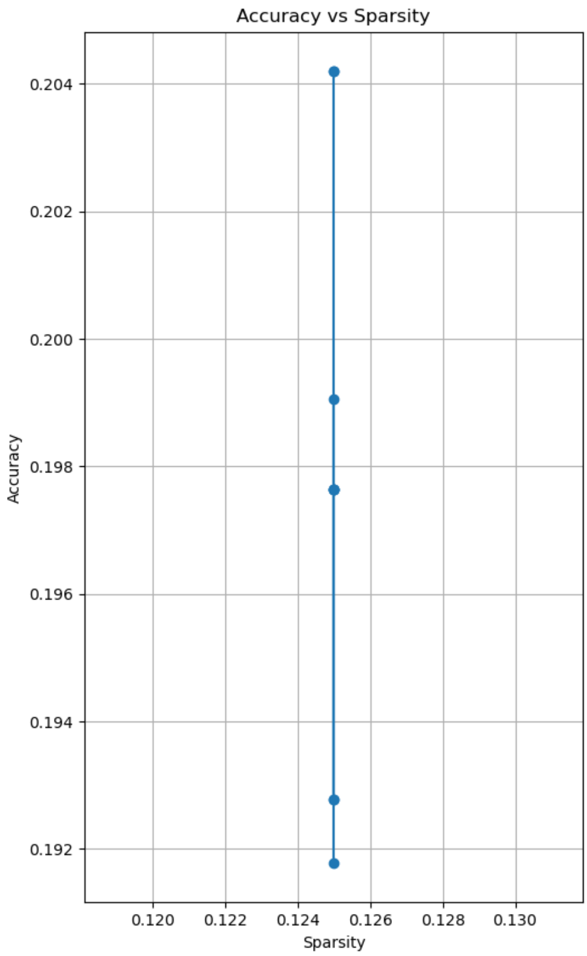

Figure 1: Pareto Front: **(1-Accuracy)** vs Sparsity