12-Week Black Box Optimization Roadmap

Complete ML Strategy with Progressive Model Complexity

Overview

Each week you get:

- **8 submissions** (one per function, 2D through 8D)
- **8 outputs** (results from black box evaluation)
- **Growing dataset** (Week N = N observations per function)

Strategy: Start simple, progressively use more sophisticated ML models as data accumulates.

Phase 1: Exploration & Foundation (Weeks 1-3)

Week 1: Initial Exploration

Data Available: 0 observations per function (starting point)

Overall Idea: Explore the search space systematically to understand scale and identify promising regions.

Models/Methods:

1. **Latin Hypercube Sampling (LHS) -** Primary method
   - Ensures good coverage of search space
   - No ML, purely sampling strategy
2. **Quasi-Random Sequences (Sobol) -** Alternative
   - Low-discrepancy sequences
   - Better space-filling than pure random

Submission Strategy:

- Test different scales: 0.1, 0.3, 0.5, 0.75, 1.0
- Use uniform values within each function
- Gather baseline data

Expected Outcome: Identify which scales/regions are promising

---

Week 2: Pattern Recognition

Data Available: 1 observation per function

Overall Idea: Exploit patterns discovered in Week 1 while continuing exploration.

Models/Methods:

1. **Pattern-Based Heuristics -** Primary method
    - Analyze Week 1 results manually
    - Exploit best-performing patterns
    - Apply transfer learning (use best pattern across functions)
2. **K-Nearest Neighbors (k=1) -** Baseline ML
    - Simplest possible ML model
    - Predicts based on single nearest neighbor
3. **Random Forest (1 tree) -** Preliminary test
    - Test if tree-based methods can handle sparse data

Submission Strategy:

- Exploit best Week 1 results (local search)
- Test if patterns transfer across functions
- Begin building ML baseline

Expected Outcome: Significant improvement on 3-4 functions

Week 3: First ML Models

Data Available: 2 observations per function

Overall Idea: Begin systematic ML with enough data for basic modeling.

Models/Methods:

1. **Gaussian Process with RBF Kernel -** Primary method
    - GP can learn with 2+ points
    - RBF kernel for smooth functions
2. **Gaussian Process with Matern Kernel -** Alternative
    - More robust for non-smooth functions
3. **Random Forest (n_estimators=10) -** Comparison
    - Ensemble baseline

4. **Bayesian Optimization with Expected Improvement (EI) -** Acquisition
    - Balance exploration/exploitation

Submission Strategy:

- Use GP + EI for automated suggestions
- Compare multiple kernel choices
- Start tracking model performance

Expected Outcome: Models begin to learn function landscapes

Phase 2: Systematic Optimization (Weeks 4-6)

Week 4: Multi-Model Ensemble

Data Available: 3 observations per function

Overall Idea: Combine multiple models for robust predictions.

Models/Methods:

1. **Gaussian Process Ensemble -** Primary method
    - Multiple kernels (RBF, Matern, Rational Quadratic)
    - Weighted combination based on validation
2. **Random Forest (n_estimators=50) -** Secondary
    - More stable with 3+ points
3. **Gradient Boosting (XGBoost) -** New addition
    - Test boosting approaches
4. **Neural Network (1 hidden layer, 10 units) -** Experimental
    - Begin testing deep learning
5. **Multiple Acquisition Functions**
    - Expected Improvement (EI)
    - Upper Confidence Bound (UCB)
    - Probability of Improvement (PI)

Submission Strategy:

- Ensemble predictions: 60% GP, 30% RF, 10% XGBoost
- Use different acquisition functions per function

- A/B test which models perform best

Expected Outcome: More stable, reliable suggestions

Week 5: Advanced Bayesian Methods

Data Available: 4 observations per function

Overall Idea: Leverage Bayesian approaches for uncertainty quantification.

Models/Methods:

1. **Multi-Task Gaussian Process -** Primary innovation
   - Share information between similar functions
   - F1-F2 (both 2D), F4-F5 (both 4D) can inform each other
2. **Thompson Sampling -** New acquisition
   - Sample from posterior distribution
   - Better exploration in later stages
3. **Gaussian Process with ARD (Automatic Relevance Determination) -** Advanced
   - Learn which dimensions are most important
4. **Bayesian Neural Network -** Experimental
   - Neural network with uncertainty quantification
5. **Ensemble with Bayesian Model Averaging -** Combination
   - Weighted ensemble based on posterior probabilities

Submission Strategy:

- Use multi-task GP to transfer knowledge between functions
- Thompson Sampling for Functions 1, 3, 7 (poor performers)
- EI/UCB for Functions 5, 8 (good performers)

Expected Outcome: Better handling of high-dimensional functions (7D, 8D)

Week 6: Dimension Reduction

Data Available: 5 observations per function

Overall Idea: Handle high-dimensional functions more effectively.

Models/Methods:

1. **Random Embedding Bayesian Optimization (REMBO) -** Primary for 6D-8D

- Projects high-D problem to low-D
- More efficient for sparse data

2. **Principal Component Analysis (PCA) + GP -** Dimension reduction
   - Find important directions in input space
3. **Sparse Gaussian Process -** Computational efficiency
   - Use inducing points for faster computation
4. **CMA-ES (Covariance Matrix Adaptation Evolution Strategy) -** Alternative
   - Evolutionary approach, good for high dimensions
5. **Trust Region Bayesian Optimization -** Local optimization
   - Focus search around best-known regions

Submission Strategy:

- REMBO for F6, F7, F8 (high-dimensional)
- Trust region for F5 (best performer)
- Standard GP for F1-F4 (lower-dimensional)

Expected Outcome: Better optimization in high dimensions

Phase 3: Refinement & Exploitation (Weeks 7-9)

Week 7: Local Search Intensification

Data Available: 6 observations per function

Overall Idea: Focus on exploiting known good regions with local models.

Models/Methods:

1. **Local Gaussian Process -** Primary method
   - Fit separate GP models around each promising region
   - Higher accuracy in local neighborhoods
2. **TuRBO (Trust Region Bayesian Optimization) -** Structured local search
   - Dynamically sized trust regions
   - State-of-the-art for later stages
3. **Gradient-Based Optimization with Surrogate -** When smooth
   - Use GP to estimate gradients

- Quasi-Newton methods (BFGS)

4. **Multi-Fidelity Optimization -** Resource allocation
   - Spend more evaluations on promising functions
   - Less on plateaued functions

Submission Strategy:

- TuRBO for functions still improving
- Local GP with small trust regions for near-optimal functions
- Pure exploitation (xi=0.001 for EI)

Expected Outcome: Fine-tuning around optimal regions

Week 8: Neural Surrogate Models
Data Available: 7 observations per function

Overall Idea: Test if deep learning can capture complex patterns.

Models/Methods:

1. **Deep Ensemble Neural Networks -** Primary deep learning
   - 5 networks with different initializations
   - Uncertainty from ensemble disagreement
2. **Neural Process -** Meta-learning approach
   - Can adapt quickly to new data
   - Good for functions with different characteristics
3. **Bayesian Neural Network with Dropout -** Uncertainty
   - MC Dropout for uncertainty estimation
4. **Graph Neural Network -** Experimental
   - Treat observations as graph nodes
   - Learn relationships between inputs
5. **Hybrid GP-NN -** Best of both worlds
   - NN for mean function, GP for residuals

Submission Strategy:

- Deep ensemble for functions with complex patterns

- GP remains baseline for comparison
- Ensemble GP + NN predictions (70-30 split)

Expected Outcome: Capture non-smooth or multi-modal functions

---

Week 9: Adaptive Strategy Selection
Data Available: 8 observations per function

Overall Idea: Automatically select best model per function based on past performance.

Models/Methods:

1. **Meta-Learning Model Selection -** Primary innovation
    - Learn which models work best for which function types
    - Predict best model based on function characteristics
2. **Automated Machine Learning (AutoML) -** Automated selection
    - Auto-sklearn or FLAML
    - Automatic hyperparameter tuning
3. **Portfolio of Models -** Hedging strategy
    - Maintain multiple models
    - Weight by recent performance
4. **Contextual Bandits -** Online learning
    - Multi-armed bandit for model selection
    - Balances exploration of models vs exploitation

Submission Strategy:

- Use meta-learner to select best model per function
- Different models for different functions
- Track and update model performance online

Expected Outcome: Optimal model selection per function

Phase 4: Advanced Techniques (Weeks 10-12)

Week 10: Transfer Learning & Meta-Learning
Data Available: 9 observations per function

Overall Idea: Leverage similarities across functions and external knowledge.

Models/Methods:

1. **Neural Process with Meta-Learning -** Primary method
   - Pre-train on synthetic functions
   - Fine-tune on your specific functions
2. **Transfer Learning from Similar Functions -** Knowledge sharing
   - Use F1 to inform F2 (both 2D)
   - Use F4 to inform F5 (both 4D)
3. **Zero-Shot Bayesian Optimization -** Prior knowledge
   - Use prior beliefs about optimization problems
4. **Warped Gaussian Process -** Flexible modeling
   - Learn transformation of output space
   - Better for functions with varying scales
5. **Multi-Output GP -** Joint modeling
   - Model all functions together
   - Capture correlations

Submission Strategy:

- Transfer learning where applicable
- Multi-output GP to share information
- Use external benchmarks as prior knowledge

Expected Outcome: Faster convergence using transferred knowledge

---

Week 11: Exploitation Focus

Data Available: 10 observations per function

Overall Idea: Pure exploitation - squeeze out final improvements.

Models/Methods:

1. **Local Quadratic Models -** High-precision local fit
   - Fit quadratic models near optimum
   - Very accurate in small regions

2. **Kriging with Nugget Effect -** Fine-grained GP
    - Very small noise parameter
    - High precision predictions
3. **Coordinate Descent -** Dimension-by-dimension
    - Optimize one dimension at a time
    - Good for near-optimal refinement
4. **Nelder-Mead with Surrogate -** Simplex method
    - Use surrogate for cheap function evaluations
5. **Line Search in Promising Directions -** Gradient-free
    - Search along directions of past improvements

Submission Strategy:

- Pure exploitation (no exploration)
- Very small perturbations around best points
- Focus on functions still showing improvement

Expected Outcome: Final precision improvements

Week 12: Final Optimization Push

Data Available: 11 observations per function

Overall Idea: Use all techniques to extract final performance gains.

Models/Methods:

1. **Ensemble of Everything -** Kitchen sink approach
    - Combine all models from previous weeks
    - Weighted by performance on validation set
2. **Hybrid Global-Local Search -** Best of both
    - Global GP for exploration (if needed)
    - Local quadratic for exploitation
3. **Stochastic Gradient Descent on Surrogate -** Fine-tuning
    - Use surrogate as differentiable approximation
    - Adam optimizer for final tweaks

4. **Particle Swarm Optimization on Surrogate -** Swarm intelligence
   - Multiple particles explore surrogate landscape

5. **Simulated Annealing -** Avoid local minima
   - Use if stuck in local optimum

6. **Multi-Start Local Optimization -** Robust optimization
   - Multiple local optimizations from best points

Submission Strategy:

- For each function, use the model with best track record
- Ensemble when uncertain
- Very small perturbations for near-optimal functions
- Larger exploration for functions still far from optimum

Expected Outcome: Maximum final performance across all functions

---

Model Summary Table

Week	Primary Model	Secondary Models	Acquisition	Focus
1	Latin Hypercube	Sobol, Random	N/A	Exploration
2	Pattern Heuristics	KNN, RF	N/A	Pattern exploit
3	GP (RBF)	GP (Matern), RF	EI	First ML
4	GP Ensemble	RF, XGBoost, NN	EI, UCB, PI	Multi-model
5	Multi-Task GP	Thompson Sampling, BNN	Thompson, EI	Transfer learning
6	REMBO	PCA+GP, CMA-ES	EI	High-dim
7	TuRBO	Local GP, Gradient	EI (xi→0)	Local search
8	Deep Ensemble	Neural Process, Hybrid	EI	Deep learning
9	Meta-Learning	AutoML, Portfolio	Adaptive	Auto-select
10	Neural Process	Transfer, Warped GP	EI	Meta-learning
11	Quadratic Local	Kriging, Coordinate	Pure exploit	Precision
12	Ensemble All	Everything	Mixed	Final push

Data Accumulation Pattern

```
Week 1:  [1] point  per function → 8 total points
```

Week 2: [2] points per function → 16 total points
Week 3: [3] points per function → 24 total points
Week 4: [4] points per function → 32 total points
Week 5: [5] points per function → 40 total points
Week 6: [6] points per function → 48 total points
Week 7: [7] points per function → 56 total points
Week 8: [8] points per function → 64 total points
Week 9: [9] points per function → 72 total points
Week 10: [10] points per function → 80 total points
Week 11: [11] points per function → 88 total points
Week 12: [12] points per function → 96 total points

Exploration vs Exploitation Schedule

Week 1-2:   🔍 🔍 🔍 🔍 🔍 90% Exploration, 10% Exploitation
Week 3-4:   🔍 🔍 🔍 🔍 🎯 70% Exploration, 30% Exploitation
Week 5-6:   🔍 🔍 🔍 🎯 🎯 50% Exploration, 50% Exploitation
Week 7-8:   🔍 🔍 🎯 🎯 🎯 30% Exploration, 70% Exploitation
Week 9-10: 🔍 🎯 🎯 🎯 🎯 10% Exploration, 90% Exploitation
Week 11-12: 🎯 🎯 🎯 🎯 🎯 5% Exploration, 95% Exploitation

Key Model Characteristics

Gaussian Process (GP)

- **Best for:** 3-8 data points, smooth functions
- **Strengths:** Uncertainty quantification, Bayesian
- **Weaknesses:** Slow with >100 points, assumes smoothness

Random Forest (RF)

- **Best for:** 5+ data points, non-smooth functions
- **Strengths:** Handles discontinuities, fast
- **Weaknesses:** No uncertainty, tends to underfit

Neural Networks (NN)

- **Best for:** 10+ data points, complex patterns
- **Strengths:** Flexible, learns any function
- **Weaknesses:** Needs more data, harder to tune

CMA-ES (Evolutionary)

- **Best for:** High dimensions (6D+), multimodal
- **Strengths:** Gradient-free, robust
- **Weaknesses:** Slower convergence, many evaluations

TuRBO (Trust Region)

- **Best for:** Later stages (7+ points), local opt
- **Strengths:** State-of-the-art for later stages
- **Weaknesses:** Needs decent starting point

Critical Decision Points

Week 3: Commit to ML or Pattern-Based?

- **If** patterns are clear and transferring → Continue pattern-based
- **If** patterns are unclear or not transferring → Full Bayesian Optimization

Week 6: Global vs Local?

- **If** still finding new optima → Continue global search (REMBO, CMA-ES)
- **If** converging to optima → Switch to local (TuRBO, local GP)

Week 9: Model Selection Strategy?

- **If** one model dominates → Use that model exclusively
- **If** different models for different functions → Use meta-learning selector

---

Expected Performance Trajectory

```
Function 5 (Best performer, 4D):
Week 1:  136.85 (initial)
Week 3:  145.0  (GP + EI)
Week 6:  152.0  (REMBO)
Week 9:  155.0  (Meta-learning)
Week 12: 157.0  (Ensemble)

Function 8 (2nd best, 8D):
Week 1:  9.54   (initial)
Week 3:  15.0   (GP)
Week 6:  22.0   (REMBO for high-dim)
```

```
Week 9:  25.0   (Deep ensemble)
Week 12: 27.0   (Final optimization)

Function 4 (Initially worst, 4D):
Week 1:  -3.99  (initial)
Week 2:  50.0   (pattern transfer from F5)
Week 4:  80.0   (GP ensemble)
Week 8:  100.0  (Neural models)
Week 12: 110.0  (Final push)
```

Implementation Priorities

Must-Have Models:

1. **Gaussian Process with EI** (Week 3-12)
2. **Random Forest** (Week 3-12, baseline)
3. **TuRBO** (Week 7-12, local optimization)

Should-Have Models:

4. **Multi-Task GP** (Week 5-12, transfer learning)
5. **Deep Ensemble** (Week 8-12, complex patterns)
6. **Meta-Learning Selector** (Week 9-12, auto-selection)

Nice-to-Have Models:

7. **REMBO** (Week 6-9, high dimensions)
8. **Neural Process** (Week 8-10, meta-learning)
9. **CMA-ES** (Week 6-8, evolutionary backup)

Success Metrics

Track these each week:

1. **Best value found per function**
2. **Average improvement per function**
3. **Model prediction accuracy** (MAE on held-out points)
4. **Exploration coverage** (% of search space sampled)
5. **Exploitation efficiency** (improvement per query near optimum)

Risk Mitigation

If Models Underperform:

- **Week 3-4:** Revert to pattern-based heuristics
- **Week 5-6:** Switch from GP to Random Forest
- **Week 7-8:** Use simpler local search (grid refinement)
- **Week 9-12:** Fall back to best-performing model from earlier weeks

If Stuck in Local Optimum:

- **Add exploration bonus** to acquisition function
- **Use CMA-ES** for 1-2 weeks to break out
- **Random restart** from unexplored regions

Documentation Requirements

Each week document:

1. **Models used** (with hyperparameters)
2. **Predictions made** (before submission)
3. **Actual results** (after submission)
4. **Model performance** (prediction error)
5. **Lessons learned** (what worked, what didn't)
6. **Next week's strategy** (based on learnings)

This creates a portfolio-ready capstone project demonstrating:
✅ Progressive complexity
✅ Multiple ML approaches
✅ Systematic experimentation
✅ Adaptive decision-making
✅ Professional documentation

Final Notes

- **Flexibility is key** - adjust based on actual results
- **Not all models will work** - that's okay, document why

- **Simple often wins -** GP + EI is hard to beat for 3-10 points
- **Your mileage may vary -** functions may have different characteristics
- **Document everything -** the journey is as important as the results

This is a roadmap, not a rigid plan. Adapt based on what you learn!