Surrogate models and active learning to enable efficient optimization

Juliane Mueller
Lawrence Berkeley National Lab
AMCR Division
JulianeMueller@lbl.gov

Exploring the frontiers in Earth system modeling with machine learning and big data
Optimization problems arise in many applications

Parametrizations in cloud simulations

Methane module in land model

Groundwater cleanup

Deep Learning

And many many more....
The objective function involves a computer simulation
- No analytic description of the objective function (black-box)
- No gradient information
- Multimodal
- The simulation is time consuming

- We cannot afford thousands of evaluations of $f(x)$
- We cannot approximate gradients

Goal is to find a solution efficiently and effectively
Surrogate models are computationally cheap approximations of the expensive to evaluate function: $f(x) = s(x) + e(x)$

- Start with an initial experimental design
Active learning tells us where to sample next

- Select a new evaluation point to run the simulation
- Use the new piece of information to update your surrogate model
Active learning tells us where to sample next

Our new evaluation

- Select a new evaluation point to run the simulation
- Use the new piece of information to update your surrogate model
Surrogate modeling and active learning strategies allow for creativity

1. Initial experimental design
2. Evaluate objective function
3. Stop? (Yes: Return best solution found; No: Create surrogate model)
4. Active learning on surrogate model to select new sample point
Polynomial models, e.g.,
\[ s(x) = a x_1 + b x_2 + c x_1 + d x_2 + e \]

Radial basis functions, e.g.,
\[ s(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|_2) + p(x) \]

Gaussian process model,
\[ s(x) = \mu + Z(x), Z(x) \sim \mathcal{N}(0, \sigma^2) \]

Active learning strategy must be modeled according to science goals

Large data settings:
Deep Learning models, e.g.,
\[ s(x) = A[\sum w_i x_i + b] \]
• Goal is to minimize an error function computed from a single column model and LES benchmark data
• 6 parameters, continuous, only bound constraints

\[
\min_{x} f(x) \\
x \in \Omega \subset \mathbb{R}^6
\]

• Previously: parameter sweep (grid search)
• Does not scale
• Oversamples uninteresting areas
• Under-samples interesting ones
• May miss “unexpected” outcomes

https://doi.org/10.1029/2018MS001449
Optimization of the Eddy-Diffusivity/Mass-Flux Shallow Cumulus and Boundary-Layer Parameterization Using Surrogate Models

• Goal is to match the black (LES) profiles
• DEF: uses default parameter values
• OPTS\(\tau\)0, OPTD are the surrogate model optimizers

Easily generalizable to many other parameter tuning problems…
Tune DL model architectures to improve performance

- How many layers do I need?
- How many nodes per layer?
- What batch size is best to use?
- What learning rate?
- ...

Hyperparameters that define the architecture → Identifying the best architecture is an iterative optimization process

- Bilevel optimization problem – hard to solve

Hyperparameters → Train the DL model → Objective function: hyperparameter performance

Select new
Tune DL model architectures to improve performance

- How many layers do I need?
- How many nodes per layer?
- What batch size is best to use?
- What learning rate?
- ...

Hyperparameters that define the architecture

Identifying the best architecture is an iterative optimization process

- Bilevel optimization problem – hard to solve

Hyperparameters → Train the DL model → Objective function: hyperparameter performance

Select new

https://doi.org/10.1007/s10898-020-00912-0
“Surrogate optimization of deep neural networks for groundwater predictions”
Need to account for DL model performance variability during architecture tuning

Model training with stochastic methods (e.g. SGD) introduces variability

- Same architecture, $\theta$
- Trained the model 3 times
- Different predictive performances

Ideally, I want to find architectures whose predictions are less impacted by this stochasticity
Assessing prediction variability requires sampling based methods

- Train each model multiple times
  - Same architecture, $\theta$
  - Trained the model 3 times
  - Different predictive performances

- Use of Monte Carlo dropout

![Diagram of DL model architecture, $\tilde{\theta}$](image)

- Use of Monte Carlo dropout
- Nested asynchronous parallelism to speed up multiple trainings

<table>
<thead>
<tr>
<th>Time</th>
<th>Quantity a.u.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial #5</td>
<td>(Data)</td>
</tr>
<tr>
<td>Trial #4</td>
<td>(Data)</td>
</tr>
<tr>
<td>Trial #17</td>
<td>(Data)</td>
</tr>
<tr>
<td>Observed</td>
<td>(Data)</td>
</tr>
</tbody>
</table>
Sample statistics provide us model performance metrics

Sample average, variance, worst case performance,…

Surrogate model: Inputs $\mapsto$ Outputs

Inputs = hyperparameters
(# layers, #nodes)

Outputs = DL model performance
(weighted sum of expected loss and uncertainty; training time)
Applicable to many different types of DL models and science apps

**Time series data:** Multilayer perceptron (MLP) to predict future groundwater levels and / or fill large gaps

![Graph showing time series data]

**3D Image reconstruction**
- Many 2D sinograms -> reconstruction algorithm to get 3D reconstructed tomogram
- Use U-Net to enable reconstruction

![Images of 3D image reconstruction]

**High energy physics applications:**
GANs to replace parametrizations

![Graph showing high energy physics applications]

**HYPPO:** A Surrogate-Based Multi-Level Parallelism Tool for Hyperparameter Optimization
• **DL models are versatile** and can be applied across applications (accelerate simulations, enable fast UQ & optimization ...)
• The UQ and architecture tuning mechanism is **generalizable to different types of DL models** (CNN, GNN, GAN, ...)

• **Surrogate model based optimization** techniques are generalizable to a wide array of parameter tuning tasks (e.g., tuning of parametrizations and tuning DL models to replace parametrizations)

• **Accelerate** training/tuning of DL models
  • Fewer training features
  • Smaller datasets: necessary vs. sufficient training dataset
  • Insensitivity of the model to small data changes may save time (no re-training, no re-tuning)

• **Quantifying uncertainty of DL model predictions**
  • Can create reliability, robustness, reproducibility -> improved trust in DL models
  • Can enable better decision making
  • Can enable wider adoption of DL models