iWet: The Intelligent WRF Ensemble Tool

Leveraging deep learning hyperparameter tuning frameworks

Derek Jensen, Donald Lucas, Clifford Anderson-Bergman, Sonia Wharton

Lawrence Livermore National Laboratory

1. Ensembles Make the World a Better Place

Ensemble methods produce more accurate predictions than individual runs and provide a measure of uncertainty

Common Types of Meteorological Ensembles

- **Initial/Boundary conditions (ICBC)**
- **Data assimilation strategy (DAS)**
- **Multi model (MM)**
- **Multi physics (MP)**
- **Stochastic physics (SP)**

Problem 1: The size of the Ensemble grows exponentially with the number of dimensions considered

e.g. $4\text{ICBC} * 2\text{DAS} * (3\text{MP})^4 = 648$ WRF Runs

Problem 2: Effective ensemble runs rely on choosing an appropriate set of input parameters

Problem 3: WRF is expensive with many discrete inputs; traditional gradient-descent optimization will not work

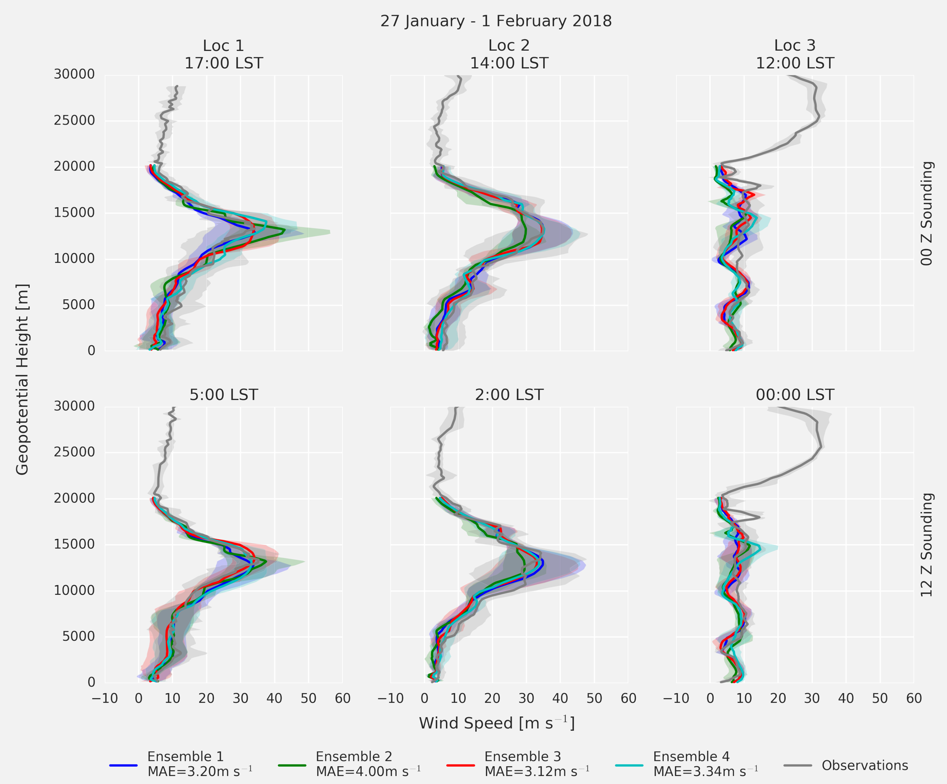

Figure: Trivial example of a four-member WRF ensemble, generated using WET, validated against radiosonde data collected at three locations from 27 January – 1 February. Lines indicate the mean values and shading represents ±1 standard deviation.

2. The WRF Ensemble Tool (WET) – Developed at LLNL

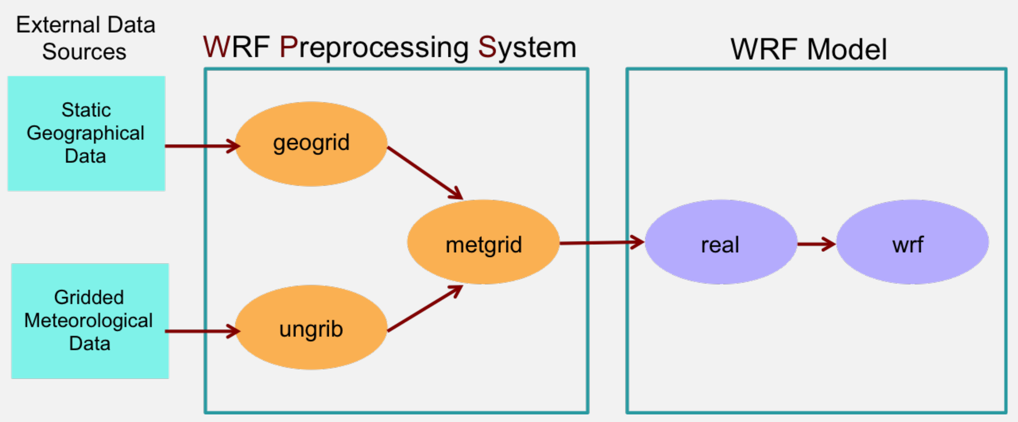

Figure: WRF model workflow. Geogrid.exe creates the terrestrial dataset, Ungrib.exe unpacks the initial and boundary condition datasets, Metgrid.exe horizontally interpolates the initial and boundary condition data onto the model domain, Real.exe vertically interpolates data to the model coordinates, and WRF.exe generates the model forecast.

WET:

- Automoatically fetches ICBC datasets
- Generates the ensemble directory structure
- Modifies the WPS and WRF namelists
- Allows for namelist consistency logic
- Executes each step of the WRF workflow in parallel
- Handles restarts

Lists define ensemble iterables:

```
WRF_NAMELIST_CHANGES = pd.Series(
  ('time_control', 'run_days'):  "0, ",
  ('time_control', 'run_hours'):  "12, ",
  ('physics', 'mp_physics'):  ['1', '2', '3', '4'],
  ('physics', 'bl_pbl_physics'):  ['1', '2'], })
```

Currently supports exhaustive and random sampling

3. Sequential Model-Based ~~hyper~~parameter Optimization (SMBO). See [1, 2]

Like a good WRF ensemble, deep neural network (DNN) performance is strongly dependent on tuning

The DNN community has developed many tools to efficiently optimize expensive, non-differentiable objective functions

SMBO builds a probability model of the DNN performance and uses it to select the most promising hyperparameters to evaluate next

Table: SMBO for DNN tuning is analogous to tuning a WRF Ensemble

Aspects of SMBO	DNN Tuning	WRF Ensemble
1. Parameter Space	n. layers, dropout, etc.	ICBC, DAS, MP, SP
2. Objective Function	DNN Training	Running WRF
3. Surrogate Model	e.g. Gaussian Process	e.g. Gaussian Process
4. Acquisition Function	e.g. Expected Improvement	e.g. Expected Improvement
5. Performance History	✓	✓

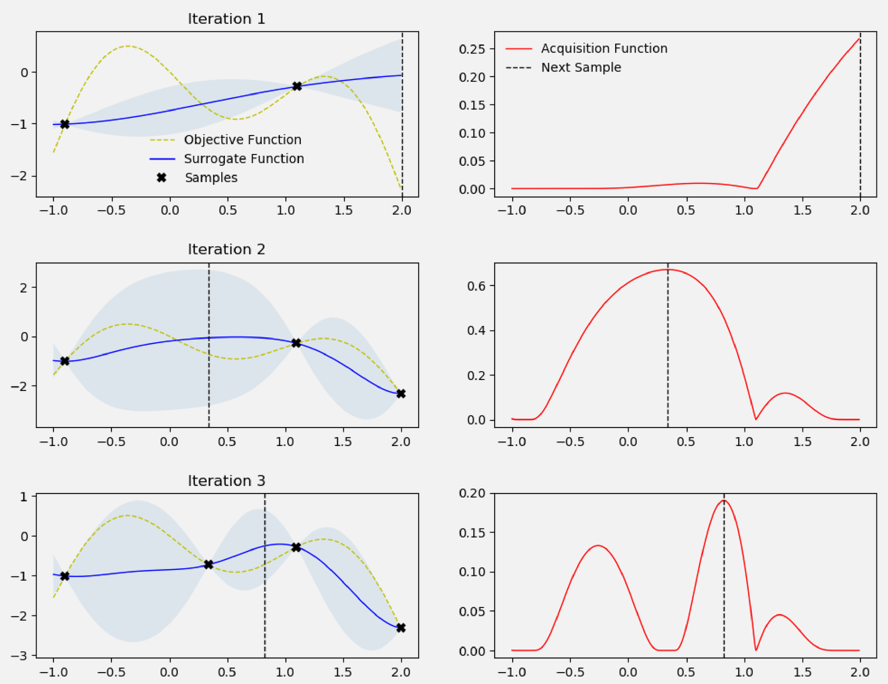

Figure: (left) Example of the unknown objective function and surrogate model and (right) acquisition function, which balances exploration (gathering more information) and exploitation (making the best decision given current information) to choose the next set of parameters [2].

4. Putting the *i* in iWet

Tune is a scalable framework for deep learning hyperparameter search, particularly SMBO

To run Tune, couple a search algorithm with a trial scheduler

Search Algorithm: Where should we sample next? e.g.

» Grid Search and Random Search
» HyperOpt: SMBO with Tree-structured Parzen Estimators

Trial Scheduler: Where/when to run the trials and when to stop them? e.g.

» Population Based Training
» Asynchronous HyperBand

A call to WET and a defined score (e.g. MSE computed between WRF output and observations) constitutes the objective function

tune

Figure: Tune is built on Ray, a flexible, high-performance distributed execution framework [3]

Feasibility Question: Can we transfer state-of-the-art methods used for tuning Deep Learning architectures, such as SMBO, to the problem of tuning ensemble parameters?

References

1. Koehrsen, Will "A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning" *Toward Data Science*, Jun 24, 2018
2. Krasser, Martin "Bayesian Optimization" *krasserm.github.io*, March 21, 2018
3. Liaw et al., "Tune: A Research Platform for Distributed Model Selection and Training" *arXiv preprint arXiv:1807.05118*, 2018

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Release Number: LLNL-POST-772377