



How to run CANDLE on Biowulf

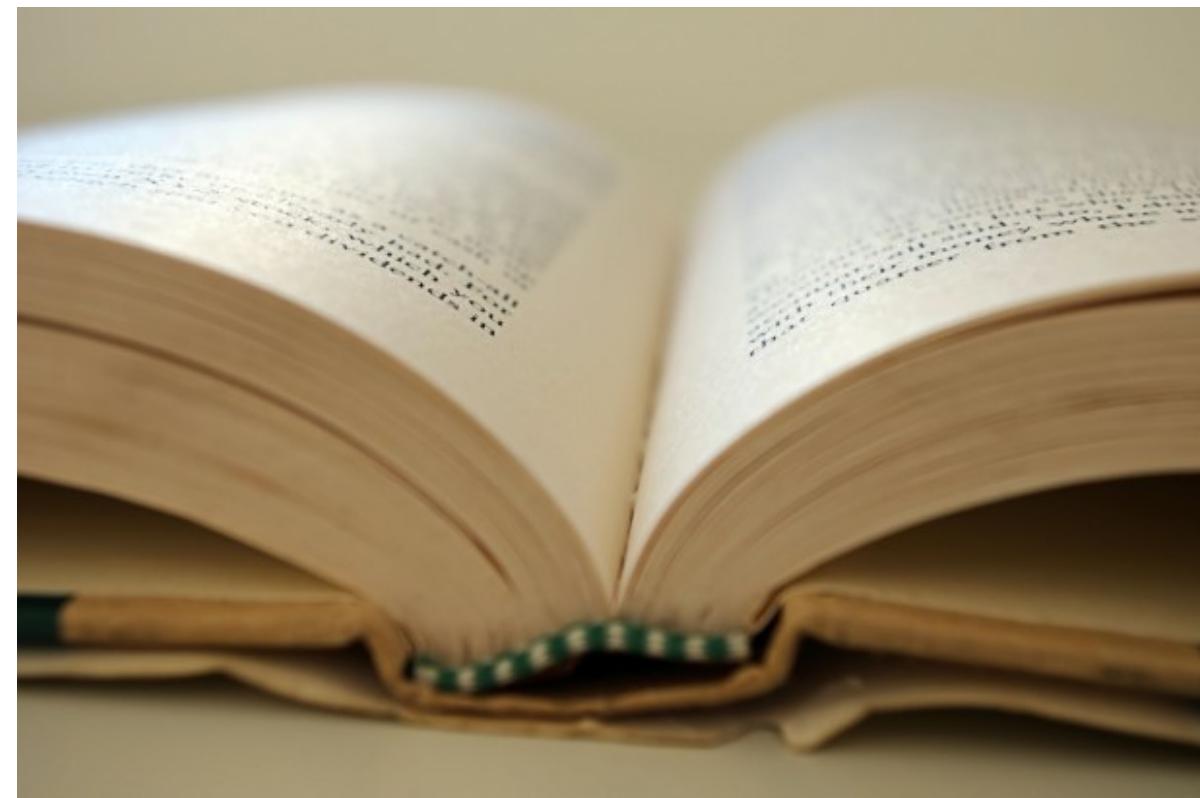
Andrew Weisman

High Performance Computing Analyst
Strategic Data Science Initiatives

July 18, 2019

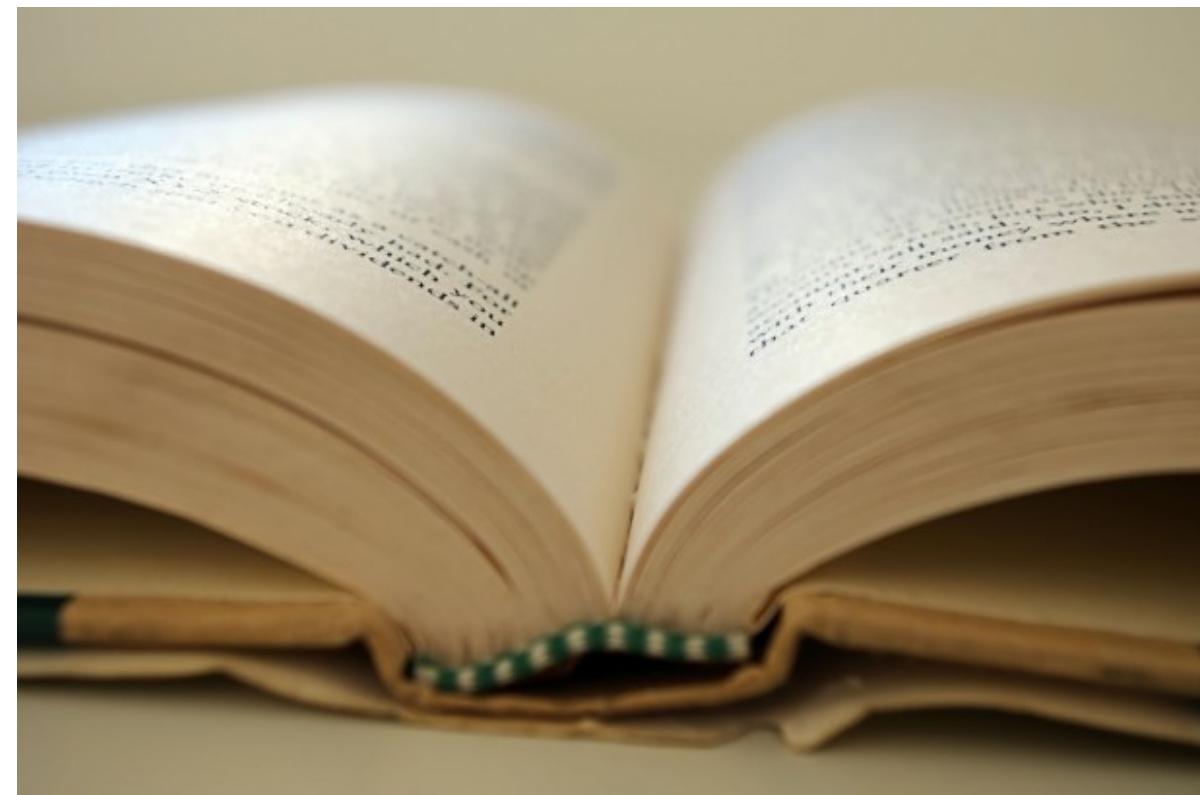
Presentation Contents

- What is CANDLE? (2 slides)
- CANDLE Quick Start (1 slide)
- How to run your model with CANDLE (7 slides)
- Postprocessing (3 slides)
- Summary (1 slide)
- Hyperparameter analysis (2 slides)



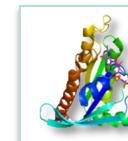
Presentation Contents

- **What is CANDLE? (2 slides)**
- CANDLE Quick Start (1 slide)
- How to run your model with CANDLE (7 slides)
- Postprocessing (3 slides)
- Summary (1 slide)
- Hyperparameter analysis (2 slides)

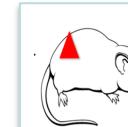


What is CANDLE?

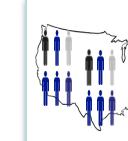
- JDACS4C
 - Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) aimed to accelerate predictive oncology at multiple levels
 - Joint effort between the DOE and the National Cancer Institute established June 2016
 - Consists of three main “pilot” efforts
- What all three pilots have in common: Need for support for HPO for deep learning analyses



Molecular Domain – Multiscale biological models
Models for RAS-RAS complex interactions
Insight into RAS related cancers (Pilot 2)



Pre-clinical Domain – Improved predictive models
Computational/hybrid predictive models of drug response
Improved experimental design (Pilot 1)



Clinical Domain – Precision oncology surveillance
Predict treatment strategies from population-scale data
Modeling patient health trajectories (Pilot 3)

What is CANDLE?

- CANDLE (CANCer Distributed Learning Environment) is an open-source software project built to provide this support
 - Actively developed
 - Scales very efficiently on the world's fastest supercomputers
- Our recent contributions:
 - We've set it up on Biowulf ("grid" and "bayesian" workflows)
 - Enabled R support (in addition to Python)
 - Made it easy to use



(Disney)

CANDLE Quick Start

- Enter a working directory on Biowulf's /data partition, e.g.,

```
mkdir /data/$USER/candle  
cd /data/$USER/candle
```

- Load the CANDLE module:

```
module load candle
```

- Copy over a CANDLE template (composed of a few files):

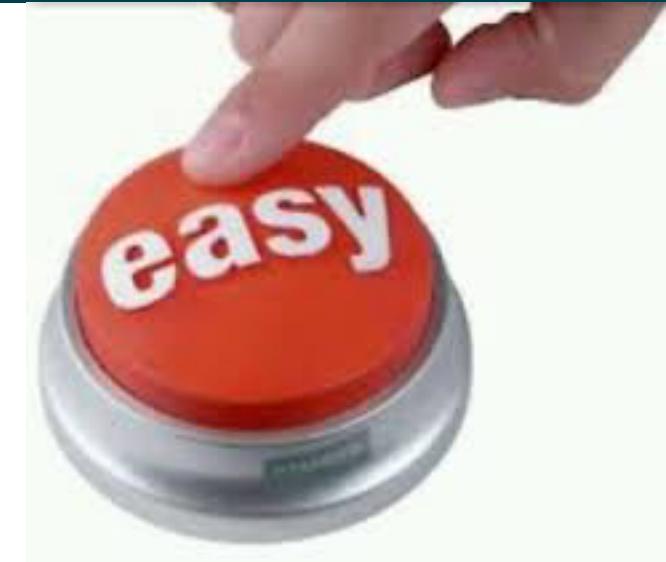
```
candle import-template {grid,bayesian,r}
```

- Submit the template:

```
candle submit-job submit_candle_job.sh
```

How do we run our own model?

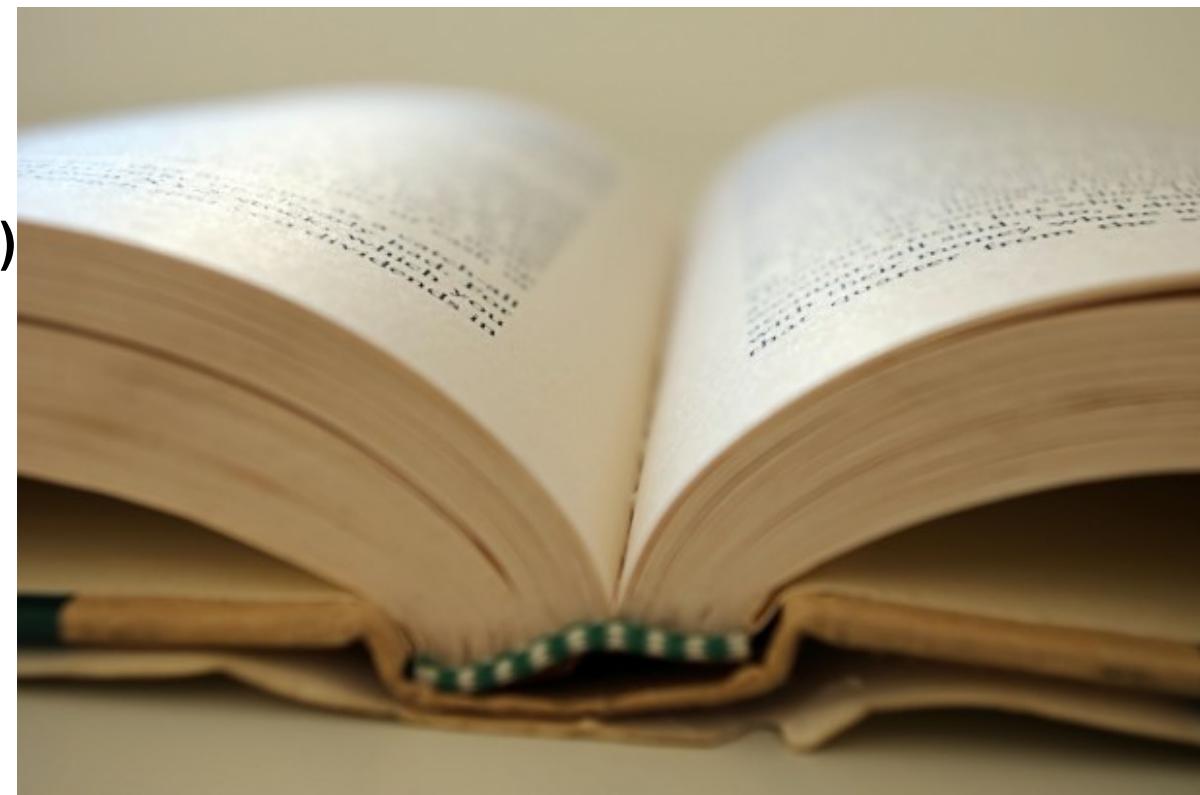
Modify these templates!



(Staples)

Presentation Contents

- What is CANDLE? (2 slides)
- CANDLE Quick Start (1 slide)
- **How to run your model with CANDLE (7 slides)**
- Postprocessing (3 slides)
- Summary (1 slide)
- Hyperparameter analysis (2 slides)



Overview of how to run CANDLE on Biowulf

- Load the CANDLE module
- Copy over a CANDLE template
- Modify your model script in two simple ways
- Place the default hyperparameters into a file
- Place the workflow hyperparameter space into a file
- Modify the CANDLE submission script
- Submit the CANDLE job



Prerequisites to implementing your own model in CANDLE

- Write a model script (e.g., “mnist.py”)
 - A “model script” is a program written in Python or R that employs a deep learning library (e.g., Keras, TensorFlow, PyTorch) to construct a machine or deep learning “model” and optimize its weights
 - Instead of a machine/deep learning model, your script can simply be any computational pipeline
- Make sure your model script runs as-is on Biowulf



How do I modify my model script for use with CANDLE?

- Decide what the hyperparameters are and substitute them with a Python dictionary or R data.frame called `hyperparams`:

Python example

Change:

```
n_conv_layers = 4  
batch_size = 128
```

To:

```
n_conv_layers = hyperparams['nconv_layers']  
batch_size = hyperparams['batch_size']
```

R example

Change:

```
n_conv_layers <- 4  
batch_size <- 128
```

To:

```
n_conv_layers <- hyperparams[["nconv_layers"]]  
batch_size <- hyperparams[["batch_size"]]
```

How do I modify my model script for use with CANDLE?

2. Return a metric (call it `val_to_return`) of how well the run performed (a single number) or a dummy number
 - Instead, for Python models you may define the variable `history` as the return value of a Keras `model.fit()` method

Python example

Add:

```
score = model.evaluate(x_test, y_test)
val_to_return = score[0]
```

Or:

```
history = model.fit(x_train, y_train, ...)
```

R example

Add:

```
val_to_return <- my_validation_loss
```

Modify the default hyperparameters file

- Place the default values of the hyperparameters into the <TEMPLATE-SPECIFIC>_default_params.txt file:

**Sample default hyperparameters file:
mnist_default_params.txt**

```
[Global Params]
epochs=20
batch_size=128
activation='relu'
optimizer='rmsprop'
num_filters=32
```



Modify the workflow settings file (grid example)

- Specify in the workflow settings file `grid_workflow-<TEMPLATE-SPECIFIC>.txt` the space of hyperparameters you'd like to run within the workflow
 - The naming scheme of this file MUST be:

`{grid,bayesian}_workflow-<ANYTHING-YOU-WANT>. {txt,R}`



Sample grid workflow settings file: `grid_workflow-mnist.txt`

```
{"id": "hpset_01", "epochs": 15, "activation": "tanh"}  
{"id": "hpset_02", "epochs": 30, "activation": "tanh"}  
{"id": "hpset_03", "epochs": 15, "activation": "relu"}  
{"id": "hpset_04", "epochs": 30, "activation": "relu"}  
{"id": "hpset_05", "epochs": 10, "batch_size": 128}  
{"id": "hpset_06", "epochs": 10, "batch_size": 256}  
{"id": "hpset_07", "epochs": 10, "batch_size": 512}
```

Modify and submit the CANDLE submission script

- Modify the submission script accordingly:

Sample submission script: submit_candle_job.sh

```
#!/bin/bash

export MODEL_SCRIPT="/data/$USER/candle/mnist.py"
export DEFAULT_PARAMS_FILE="/data/$USER/candle/mnist_default_params.txt"
export WORKFLOW_SETTINGS_FILE="/data/$USER/candle/grid_workflow-mnist.txt"
export NGPUS=7
export GPU_TYPE="k80"
export WALLTIME="00:45:00"

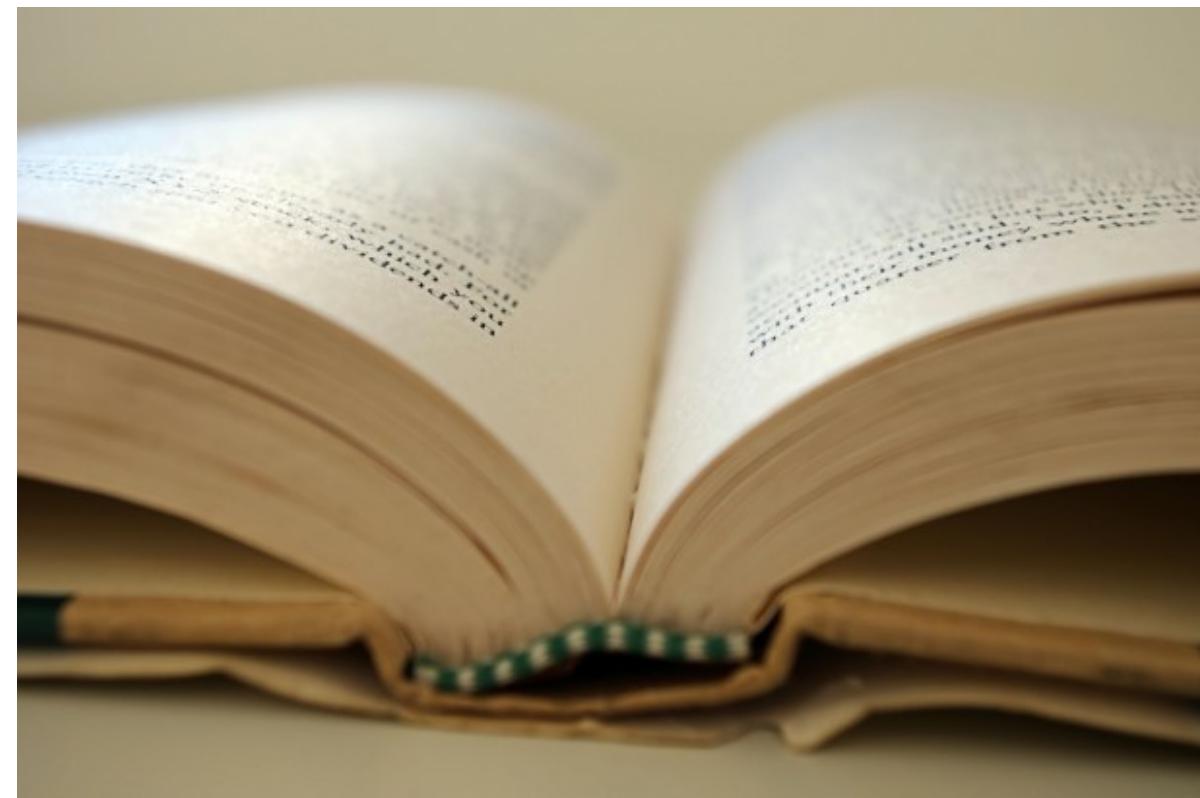
##### MODIFY ONLY ABOVE; DO NOT MODIFY BELOW #####
$CANDLE_WRAPPERS/templates/scripts/run_workflows.sh
```

- Submit the job:

```
candle submit-job submit_candle_job.sh
```

Presentation Contents

- What is CANDLE? (2 slides)
- CANDLE Quick Start (1 slide)
- How to run your model with CANDLE (7 slides)
- **Postprocessing (3 slides)**
- Summary (1 slide)
- Hyperparameter analysis (2 slides)



Sample directory structure after CANDLE job completion

```
.  
├── experiments  
│   └── x000  
│       ├── cfg-sys-biowulf.sh  
│       ├── grid_workflow-mnist.txt  
│       ├── jobid.txt  
│       ├── metadata.json  
│       ├── output.txt  
│       └── run  
│           ├── hpset_01  
│           ├── hpset_02  
│           ├── hpset_03  
│           ├── hpset_04  
│           ├── hpset_05  
│           ├── hpset_06  
│           └── hpset_07  
│           ├── submit.sh  
│           ├── turbine.log  
│           ├── turbine-slurm.sh  
│           └── workflow.sh.log  
└── workflow.tic  
└── last-exp -> /gpfs/gsfs10/users/weismanal/notebook/2019-07-11/test_grid/experiments/X000  
└── submit_candle_job.sh
```



Aggregating the results

- Put the hyperparameters and corresponding metrics into CSV format:

```
candle aggregate-results  
<EXPERIMENT-DIRECTORY>
```

- Example:

```
candle aggregate-results  
/data/$USER/candle/experiments  
/X004
```

- This will print to the file `candle_results.csv` all the hyperparameters and metrics in CSV format



Sample CSV file in Microsoft Excel

	A	B	C	D	E	F	G
1	result	dirname	id	batch_size	num_cores	use_gpu	learning_rate
2	73.26	hpset_00420	hpset_00420	1024	14	TRUE	1000
3	73.778	hpset_00415	hpset_00415	1024	14	TRUE	150
4	73.852	hpset_00416	hpset_00416	1024	14	TRUE	200
5	74.196	hpset_00401	hpset_00401	1024	8	TRUE	150
6	74.365	hpset_00418	hpset_00418	1024	14	TRUE	500
7	74.469	hpset_00414	hpset_00414	1024	14	TRUE	100
8	74.631	hpset_00404	hpset_00404	1024	8	TRUE	500
9	74.711	hpset_00417	hpset_00417	1024	14	TRUE	300
10	74.917	hpset_00400	hpset_00400	1024	8	TRUE	100
11	74.972	hpset_00406	hpset_00406	1024	8	TRUE	1000
12	75.005	hpset_00403	hpset_00403	1024	8	TRUE	300
13	75.124	hpset_00402	hpset_00402	1024	8	TRUE	200
14	75.43	hpset_00348	hpset_00348	512	14	TRUE	500
15	75.525	hpset_00350	hpset_00350	512	14	TRUE	1000
16	75.589	hpset_00419	hpset_00419	1024	14	TRUE	700
17	75.822	hpset_00349	hpset_00349	512	14	TRUE	700

Summary of how to run CANDLE on Biowulf

- Enter working directory on Biowulf's /data partition, e.g., `cd /data/$USER/candle`
- Load the CANDLE module: `module load candle`
- Copy over one of the CANDLE templates:
`candle import-template {grid,bayesian,r}`
- Ensure your model script works standalone on Biowulf
- Modify your model script in two ways:
 1. Substitute the hyperparameters with a `hyperparams` dictionary/`data.frame`
 2. Return a single number metric of performance in `val_to_return`
- Set the default hyperparameters in the `XXXX_default_params.txt` file
- Set the workflow hyperparameter space in `{grid,bayesian}_workflow-YYYY.{txt,R}`
- Modify the CANDLE submission script `submit_candle_job.sh` accordingly
- Submit the CANDLE job: `candle submit-job submit_candle_job.sh`
- Aggregate the results to the file `candle_results.csv`: `candle aggregate-results <EXPERIMENT-DIRECTORY>`

Presentation Contents

- What is CANDLE? (2 slides)
- CANDLE Quick Start (1 slide)
- How to run your model with CANDLE (7 slides)
- Postprocessing (3 slides)
- Summary (1 slide)
- **Hyperparameter analysis (2 slides)**



Sample hyperparameter analysis: (2) Split up into GPU vs. CPU only

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	result	dirname	id	batch_size	num_cores	use_gpu	learning_rate			result	dirname	id	batch_size	num_cores	use_gpu	learning_rate
2																
3	73.26	hpset_00420	hpset_00420	1024	14	TRUE	1000			104.3	hpset_00343	hpset_00343	512	14	FALSE	1000
4	73.778	hpset_00415	hpset_00415	1024	14	TRUE	150			104.873	hpset_00341	hpset_00341	512	14	FALSE	500
5	73.852	hpset_00416	hpset_00416	1024	14	TRUE	200			105.224	hpset_00339	hpset_00339	512	14	FALSE	200
6	74.196	hpset_00401	hpset_00401	1024	8	TRUE	150			105.254	hpset_00337	hpset_00337	512	14	FALSE	100
7	74.365	hpset_00418	hpset_00418	1024	14	TRUE	500			105.353	hpset_00342	hpset_00342	512	14	FALSE	700
8	74.469	hpset_00414	hpset_00414	1024	14	TRUE	100			105.71	hpset_00338	hpset_00338	512	14	FALSE	150
9	74.631	hpset_00404	hpset_00404	1024	8	TRUE	500			105.879	hpset_00408	hpset_00408	1024	14	FALSE	150
10	74.711	hpset_00417	hpset_00417	1024	14	TRUE	300			106.061	hpset_00413	hpset_00413	1024	14	FALSE	1000
11	74.917	hpset_00400	hpset_00400	1024	8	TRUE	100			106.09	hpset_00340	hpset_00340	512	14	FALSE	300
12	74.972	hpset_00406	hpset_00406	1024	8	TRUE	1000			106.843	hpset_00411	hpset_00411	1024	14	FALSE	500
13	75.005	hpset_00403	hpset_00403	1024	8	TRUE	300			107.166	hpset_00412	hpset_00412	1024	14	FALSE	700
14	75.124	hpset_00402	hpset_00402	1024	8	TRUE	200			107.46	hpset_00409	hpset_00409	1024	14	FALSE	200
15	75.43	hpset_00348	hpset_00348	512	14	TRUE	500			108.253	hpset_00323	hpset_00323	512	8	FALSE	100
16	75.525	hpset_00350	hpset_00350	512	14	TRUE	1000			108.877	hpset_00407	hpset_00407	1024	14	FALSE	100
17	75.589	hpset_00419	hpset_00419	1024	14	TRUE	700			110.057	hpset_00410	hpset_00410	1024	14	FALSE	300
18	75.822	hpset_00349	hpset_00349	512	14	TRUE	700			110.139	hpset_00268	hpset_00268	256	14	FALSE	150
19	75.833	hpset_00344	hpset_00344	512	14	TRUE	100			110.147	hpset_00273	hpset_00273	256	14	FALSE	1000
20	75.946	hpset_00345	hpset_00345	512	14	TRUE	150			110.207	hpset_00329	hpset_00329	512	8	FALSE	1000
21	76.048	hpset_00405	hpset_00405	1024	8	TRUE	700			110.332	hpset_00267	hpset_00267	256	14	FALSE	100
22	76.274	hpset_00346	hpset_00346	512	14	TRUE	200			110.341	hpset_00271	hpset_00271	256	14	FALSE	500

Sample hyperparameter analysis: (3) Run Pearson correlation

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	result	dirname	id	batch_size	num_cores	use_gpu	learning_rate			result	dirname	id	batch_size	num_cores	use_gpu	learning_rate
2	1			-0.65	-0.10		0.00			1			-0.21	-0.66		0.00
3	73.26	hpset_00420	hpset_00420	1024	14	TRUE	1000			104.3	hpset_00343	hpset_00343	512	14	FALSE	1000
4	73.778	hpset_00415	hpset_00415	1024	14	TRUE	150			104.873	hpset_00341	hpset_00341	512	14	FALSE	500
5	73.852	hpset_00416	hpset_00416	1024	14	TRUE	200			105.224	hpset_00339	hpset_00339	512	14	FALSE	200
6	74.196	hpset_00401	hpset_00401	1024	8	TRUE	150			105.254	hpset_00337	hpset_00337	512	14	FALSE	100
7	74.365	hpset_00418	hpset_00418	1024	14	TRUE	500			105.353	hpset_00342	hpset_00342	512	14	FALSE	700
8	74.469	hpset_00414	hpset_00414	1024	14	TRUE	100			105.71	hpset_00338	hpset_00338	512	14	FALSE	150
9	74.631	hpset_00404	hpset_00404	1024	8	TRUE	500			105.879	hpset_00408	hpset_00408	1024	14	FALSE	150
10	74.711	hpset_00417	hpset_00417	1024	14	TRUE	300			106.061	hpset_00413	hpset_00413	1024	14	FALSE	1000
11	74.917	hpset_00400	hpset_00400	1024	8	TRUE	100			106.09	hpset_00340	hpset_00340	512	14	FALSE	300
12	74.972	hpset_00406	hpset_00406	1024	8	TRUE	1000			106.843	hpset_00411	hpset_00411	1024	14	FALSE	500
13	75.005	hpset_00403	hpset_00403	1024	8	TRUE	300			107.166	hpset_00412	hpset_00412	1024	14	FALSE	700
14	75.124	hpset_00402	hpset_00402	1024	8	TRUE	200			107.46	hpset_00409	hpset_00409	1024	14	FALSE	200
15	75.43	hpset_00348	hpset_00348	512	14	TRUE	500			108.253	hpset_00323	hpset_00323	512	8	FALSE	100
16	75.525	hpset_00350	hpset_00350	512	14	TRUE	1000			108.877	hpset_00407	hpset_00407	1024	14	FALSE	100
17	75.589	hpset_00419	hpset_00419	1024	14	TRUE	700			110.057	hpset_00410	hpset_00410	1024	14	FALSE	300
18	75.822	hpset_00349	hpset_00349	512	14	TRUE	700			110.139	hpset_00268	hpset_00268	256	14	FALSE	150
19	75.833	hpset_00344	hpset_00344	512	14	TRUE	100			110.147	hpset_00273	hpset_00273	256	14	FALSE	1000
20	75.946	hpset_00345	hpset_00345	512	14	TRUE	150			110.207	hpset_00329	hpset_00329	512	8	FALSE	1000
21	76.048	hpset_00405	hpset_00405	1024	8	TRUE	700			110.332	hpset_00267	hpset_00267	256	14	FALSE	100
22	76.274	hpset_00346	hpset_00346	512	14	TRUE	200			110.341	hpset_00271	hpset_00271	256	14	FALSE	500

Questions?

How to generate the grid of hyperparameters automatically

- You can generate a grid of hyperparameters automatically by running

```
candle generate-grid <PYTHON-LIST-1> <PYTHON-LIST-2> ...
```

- For example, running

```
candle generate-grid "[{'nlayers': np.arange(5,15,2)}]" "[{'dir': ['x', 'y', 'z']}]"
```

would output to the file `grid_workflow-XXXX.txt`:

```
{"id": "hpset_00001", "nlayers": 5, "dir": "x"}  
{"id": "hpset_00002", "nlayers": 5, "dir": "y"}  
{"id": "hpset_00003", "nlayers": 5, "dir": "z"}  
{"id": "hpset_00004", "nlayers": 7, "dir": "x"}  
{"id": "hpset_00005", "nlayers": 7, "dir": "y"}  
{"id": "hpset_00006", "nlayers": 7, "dir": "z"}  
{"id": "hpset_00007", "nlayers": 9, "dir": "x"}
```

etc.

Modify the workflow settings file (bayesian example)

- Specify in the workflow settings file bayesian_workflow-<TEMPLATE-SPECIFIC>.txt the space of hyperparameters you'd like to run within the workflow:

Sample bayesian workflow settings file: bayesian_workflow-mnist.R

```
param.set <- makeParamSet(  
  makeDiscreteParam("batch_size", values = c(16, 32)),  
  makeIntegerParam("epochs", lower = 2, upper = 5),  
  makeDiscreteParam("activation", values = c("softmax", "softplus", "relu")),  
  makeNumericParam("drop", lower = 0, upper = 0.9)  
)
```

Section Title

Presenter



Presentation Title

Presenter

Title, Affiliation

Date 1, 2018

Secondary Slide Title

- Bullet point
- Bullet point
- Bullet point