Argonne NATIONAL LABORATORY

MIGRATING YOUR ML TO CANDLE

HARRY YOO
Software Engineer
Data Science and Learning Division

hsyoo@anl.gov
Mon, Jan 14th 2019,
ECP Annual Meeting

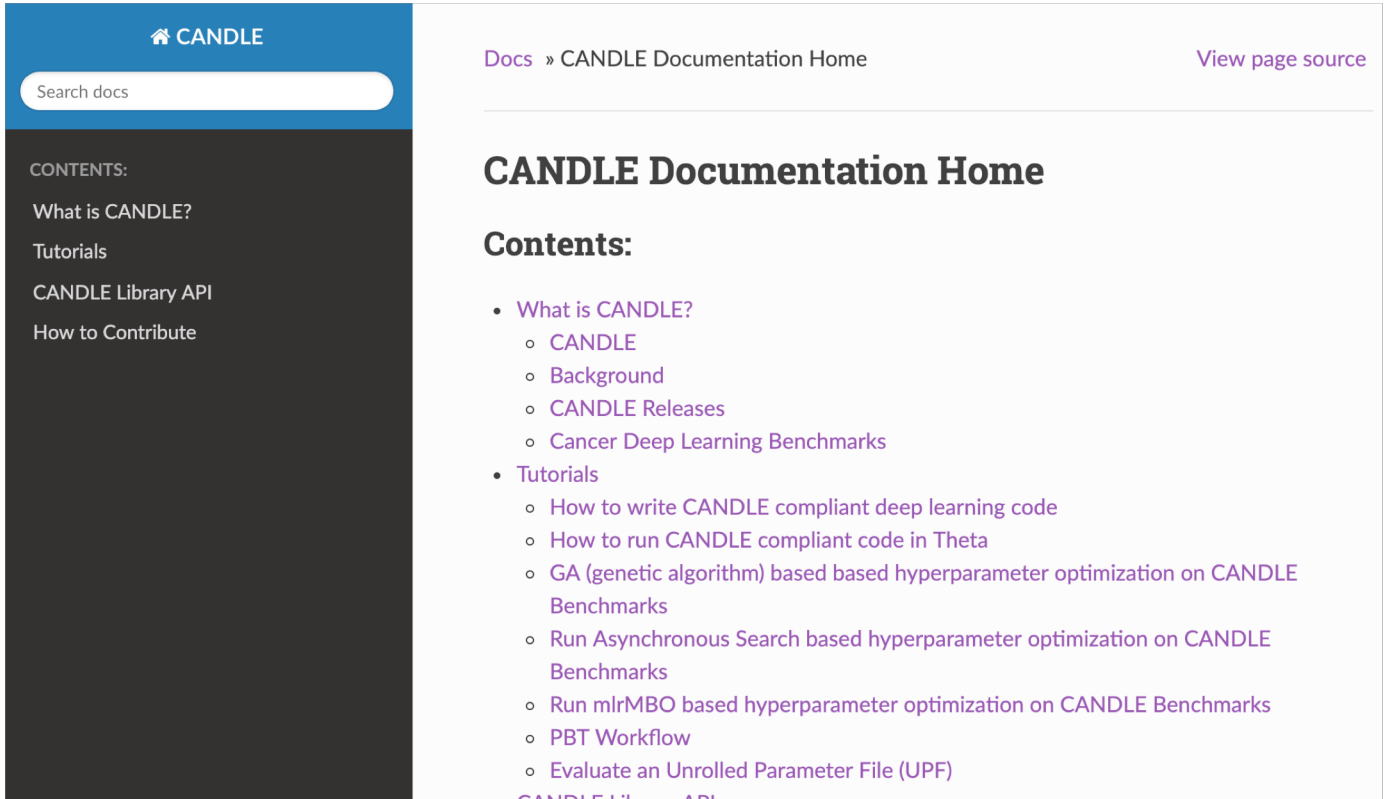
Argonne NATIONAL LABORATORY

AGENDA

- Documentation Site
- Code Structure
- Parameters
- Custom data loading

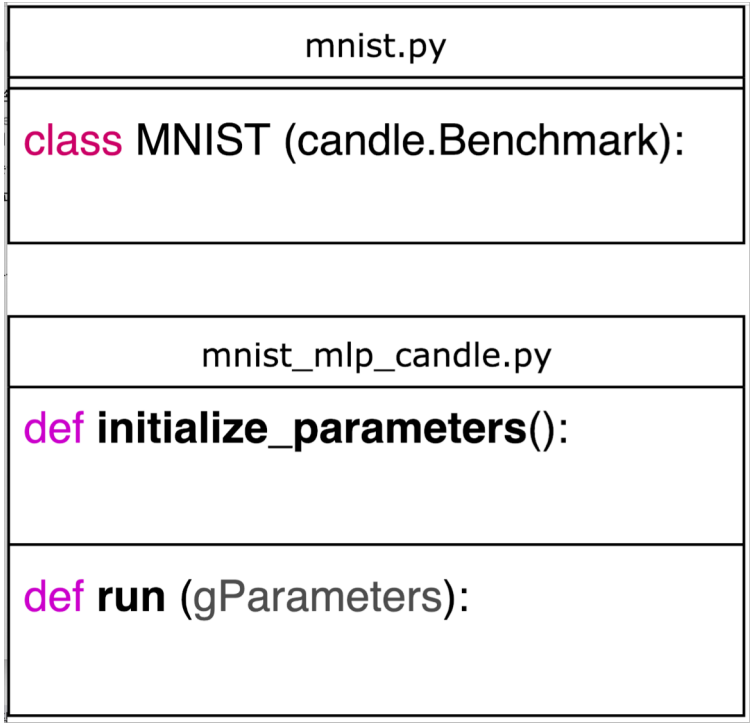
DOCUMENTATION

https://ecp-candle.github.io/Candle/



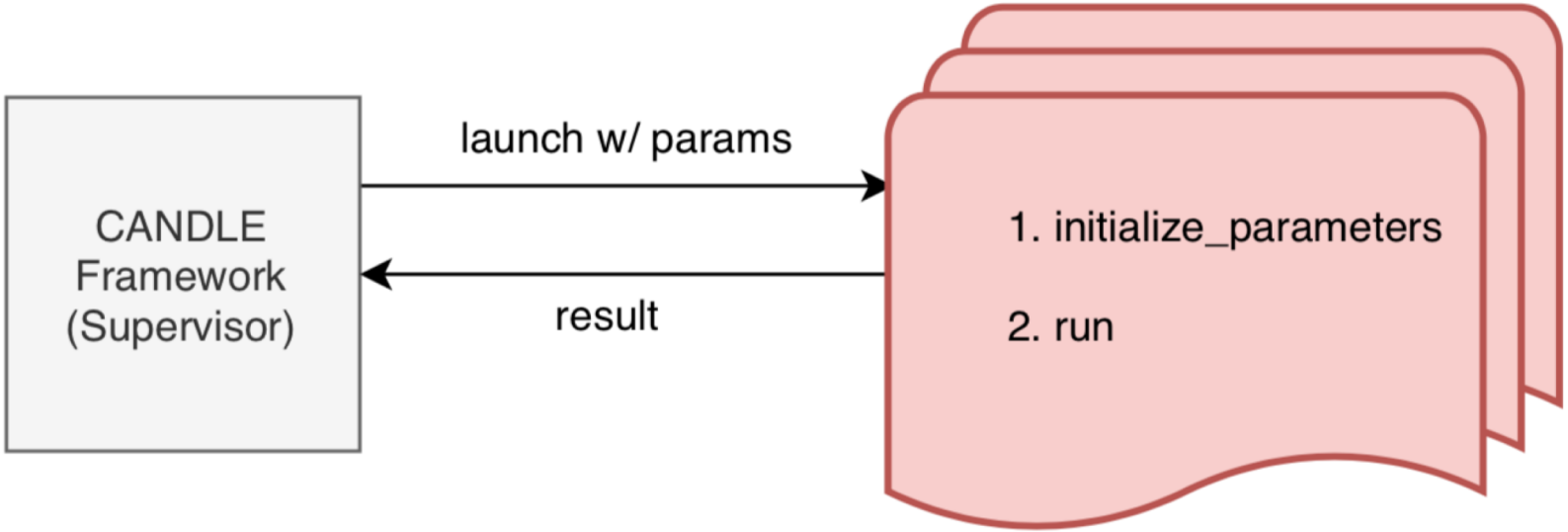
CODE STRUCTURE

1. `import candle_keras as candle`
2. Construct a class extending `candle.Benchmark`
3. Implement
   - `initialize_parameters`
   - `run`
4. (later) implement data loading for your data



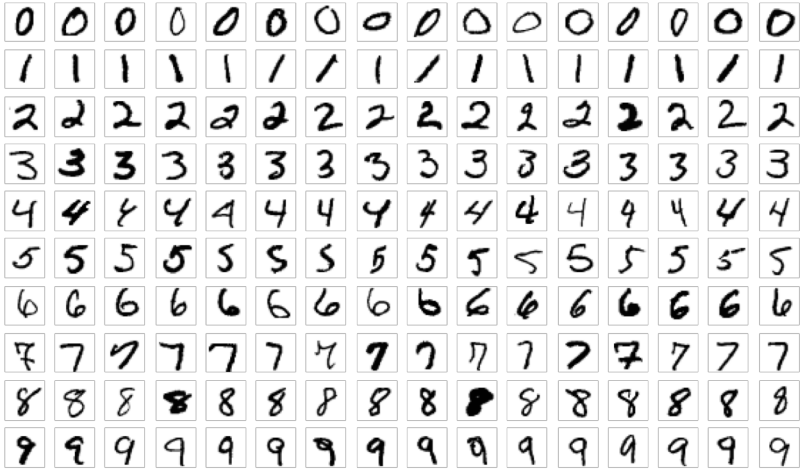
`https://github.com/ECP-CANDLE/Candle/tree/master/examples/mnist`

WHY



ABOUT MNIST

- Handwritten digits (0 – 9)
- 28 x 28 pixels
- 60,000 training, 10,000 test samples
- http://yann.lecun.com/exdb/mnist/
- `from keras.datasets import mnist`



MNIST MLP

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               401920
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                5130
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
```

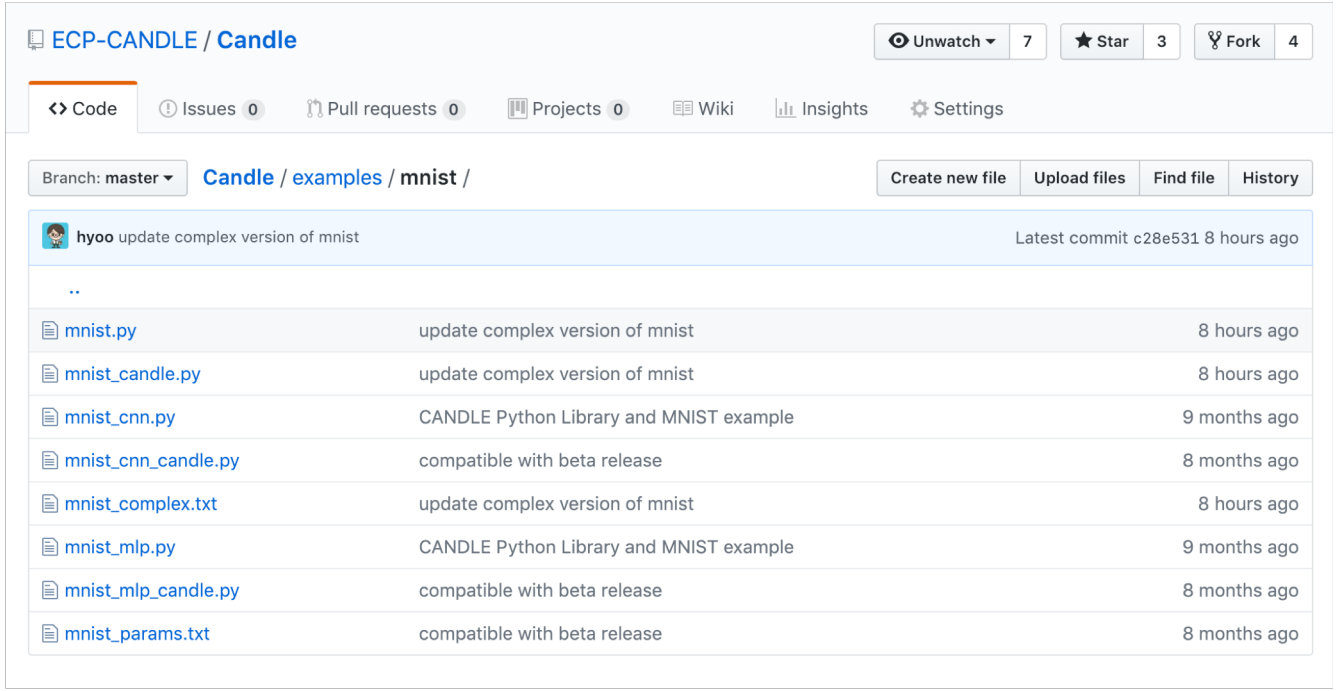
https://github.com/ECP-CANDLE/Candle/blob/master/examples/mnist/mnist_mlp.py

MNIST CNN

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 9216)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               1179776
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 1,199,882
Trainable params: 1,199,882
Non-trainable params: 0
```

MNIST CODE REVIEW

https://github.com/ECP-CANDLE/Candle/tree/master/examples/mnist



HANDS ON

```
(keras) ❯ python mnist_mlp_candle.py
Using TensorFlow backend.
Configuration file:  /Users/hsyoo/dev/CANDLE/Candle/examples/mnist/mnist_params.txt
{'activation': 'relu', 'batch_size': 128, 'epochs': 20, 'optimizer': 'rmsprop'}
Params:
{'activation': 'relu',
 'batch_size': 128,
 'datatype': <class 'numpy.float32'>,
 'epochs': 20,
 'experiment_id': 'EXP000',
 'gpus': [],
 'logfile': None,
 'optimizer': 'rmsprop',
 'output_dir': '/Users/hsyoo/dev/CANDLE/Candle/examples/mnist/Output/EXP000/RUN000',
 'rng_seed': 7102,
 'run_id': 'RUN000',
 'shuffle': False,
 'timeout': -1,
 'train_bool': True,
 'verbose': None}
```

PARAMETERS

- Default Parameters
  - Use `-help` to check
- Add / mandate parameters
- Use model parameter file to set default sets
  - `mnist_params.txt`
- Overwrite parameters as needed

OVERWRITING PARAMETERS

1. Default value defined in CANDLE Library
2. Values defined in default parameter file. e.g. `mnist_params.txt`
3. Parameters file overwritten by `--conf`
4. Individual parameter values provided by argument. e.g. `-e 1`

HANDS ON

- Run without arguments
  - `python mnist_mlp_candle.py`
- Run using model file
  - `python mnist_mlp_candle.py –conf model_complex.txt`
- Run overwriting with argument
  - `python minst_mlp_candle.py –e 1`
- Let make it more complicated
  - `python mnist_candle.py`

ADDITIONAL PARAMS

- Declare additional parameters in JSON format

```
additional_definitions = [
{'name':'latent_dim',
    'action':'store',
    'type': int,
    'help':'latent dimensions'},
{'name':'model',
    'default':'ae',
    'choices':['ae', 'vae', 'cvae'],
    'help':'model to use: ae, vae, cvae'},
{'name':'use_landmark_genes',
    'type': candle.str2bool,
    'default': False,
    'help':'use the 978 landmark genes from LINCS (L1000) as expression features'},
```

https://github.com/ECP-CANDLE/Benchmarks/blob/master/Pilot1/P1B1/p1b1.py

REQUIRED PARAMS

```
74  required = [
75      'activation',
76      'batch_size',
77      'dense',
78      'drop',
79      'epochs',
80      'initialization',
81      'learning_rate',
82      'loss',
83      'noise_factor',
84      'optimizer',
85      'rng_seed',
86      'model',
87      'scaling',
88      'validation_split',
```

https://github.com/ECP-CANDLE/Benchmarks/blob/master/Pilot1/P1B1/p1b1.py

LOADING DATA

```
def load_data(params, seed):
    drop_cols = ['case_id']
    onehot_cols = ['cancer_type']
    y_cols = ['cancer_type']

    if params['use_landmark_genes']:
        lincs_file = 'lincs1000.tsv'
        lincs_path = candle.fetch_file(params['url_p1b1'] + lincs_file, 'Pilot1')
        df_l1000 = pd.read_csv(lincs_path, sep='\t')
        x_cols = df_l1000['gdc'].tolist()
        drop_cols = None
    else:
        x_cols = None

    train_path = candle.fetch_file(params['url_p1b1'] + params['file_train'], 'Pilot1')
    test_path = candle.fetch_file(params['url_p1b1'] + params['file_test'], 'Pilot1')

    return candle.load_csv_data(train_path, test_path,
                                x_cols=x_cols,
                                y_cols=y_cols,
                                drop_cols=drop_cols,
                                onehot_cols=onehot_cols,
                                n_cols=params['feature_subsample'],
                                shuffle=params['shuffle'],
                                scaling=params['scaling'],
                                dtype=params['datatype'],
                                validation_split=params['validation_split'],
                                return_dataframe=False,
                                return_header=True,
                                seed=seed)
```

Check CANDLE Library API for more detail

THANKS