

## Command to run CLI

```
python run.py --infile <input filename> --outfile <output filename> --learnrate <(0.0,1.0]>
```

Other command-line options are given below.

## Configuration parameters as command-line options

- Specify the learning rate (default value is 0.01) -- *\*required*.

`--learnrate`

*Determines the step size of the weight update and is critical to the convergence of the algorithm to the global minima of the objective function. A very high learning rate might end up missing certain local minima. A very low learning will end up slowing down the learning process.*

- Specify the number of layers in the neural network.

`--nlayers`

*Number of layers depends on the number of features that need to be extracted from the data.*

- Specify the dimensions of each layer in the neural network.

`--layerdim`

*Number of computation units in each layer.*

- Specify the fraction of RUnits that are dropped out (value is in range [0.0, 1.0]).

`--dropout_fraction_ru`

*Drop a fraction of recurrent connections that remember the past results. Dropping RUnits would reduce exposure to data and hence, prevent overfitting.*

- Specify the fraction of input units that are dropped out (value is in range [0.0, 1.0]).

`--dropout_fraction_rw`

*Drop fraction of inputs from passing to the next layer in the network. This helps in being more robust to noise.*

- Specify the optimizer.

`--optimizer`

*There are currently three optimizers (Adam, SGD and RMSprop).*

- Specify the momentum.

`--momentum`

*Weighted sum of past gradients that is used to accelerate learning and provide direction to the optimiser.*

- Specify the training percent (The value is in range (0.0, 1.0]).

`--trainpct`

*Percent of data to be used for training. The remaining would be used for testing and evaluation.*

- Specify the error metric.

`--errmetric`

- Specify the number of epochs.

`--epoch`

*Number of passes through the whole training data.*

## Command-line options

---

- Specify input filename (.csv) -- *\*required*.

`--infile`

*The input file currently needs to contain 1-dimensional data.*

- Specify output filename -- *\*required*.

`--outfile`

- Specify a config JSON file as input.

`--config`

*Can use this to provide a file containing a JSON with appropriate parameters as the configuration to run the neural network. If config file provided, then all configuration parameters specified (those specified above) on the command-line would be ignored.*

*Example config JSON is shown below,*

```
{
  "n_layers": 4,
  "dropout_fraction_ru": 0.1,
  "dropout_fraction_rw": 0.1,
  "layer_dimensions": [1, 60, 60, 1],
  "optimizer": "adam",
  "learning_rate": 0.001,
  "momentum": 0.1,
  "training_percent": 0.5,
  "err_metric": "mean_squared_error",
```

```
"epoch": 10
}
```

- Specify log filename (default logfile is \_log).

```
--logfile
```

- Append the run configuration to the logfile.

```
--append
```

## Command to run GUI

```
python gui.py
```

## GUI Guide -- Configuration Input

The screenshot shows the 'LSTM tool v0.1.1' GUI. It features a 'PARAMETER(S):' section with various input fields and buttons. Red arrows point from yellow text annotations to specific GUI elements. A blue arrow points from a yellow annotation to the 'Run' button.

**Annotations and their corresponding GUI elements:**

- Browse to load input filename**: Points to the 'Load input filename (\*)' browse button.
- Specify output filename**: Points to the 'Save output filename as (\*)' browse button.
- Delete uploaded config file**: Points to the 'Delete Config' button.
- upload config file**: Points to the 'Browse' button under 'Use config file:'.
- Percentage of input data used for training**: Points to the 'Training percentage' input field (0.7).
- Number of layers in the neural network**: Points to the 'Number of layers' input field (3).
- Number of passes through the whole training data**: Points to the 'Epoch' input field (100).
- Select an optimizer from the menu -- {adam, SGD, RMSprop, nadam}**: Points to the 'Optimizer type' dropdown menu (adam).
- Choose a metric to measure error of the objective function -- {MSE}**: Points to the 'Error Metric' dropdown menu (MSE).
- Specify the no. of computation units per layer**: Points to the 'Layer dimensions [LAYERDIM1,LAYERDIM2,...]' input field (1,4,1).
- Number that controls the step size of the weight update**: Points to the 'Learning rate (\*)' input field.
- Weighted sum of past gradients**: Points to the 'Momentum' input field (0.0).
- Save the config as JSON which can loaded in future**: Points to the 'Save Config' button.
- Now build, train and evaluate model !**: Points to the 'Run' button.

Other GUI elements include: 'All fields marked with (\*) are required', 'Upload config JSON file or fill the configurations below', 'Append run configuration to logfile' checkbox, and a 'Delete Result Buffer' button at the bottom right.

## GUI Guide -- Results

PARAMETER(S) :

All fields marked with (\*) are required

Upload config JSON file or fill the configurations below

Load input filename (\*) :

/Users/abhishekjain/nnet/input\_files/exchange-rate-twi-may-1970-aug-1.csv

Save output filename as (\*) :

/Users/abhishekjain/nnet/output

Use config file :

Training percentage :

Number of layers :

Epoch :

Fraction of R\_Units to be dropped [0.0, 1.0) :

Fraction of input units to be dropped [0.0, 1.0) :

Optimizer type :

Error Metric :

Layer dimensions [LAYERDIM1,LAYERDIM2,...] :

Learning rate (\*) :

Momentum :

Append run configuration to logfile ☒

Save config :

Developed by Abhishek Jain, Pradyumna Kaushik and Sreedhar Kumar

LSTM tool v0.1.1

Using TensorFlow backend.

Train on 105 samples, validate on 45 samples

Epoch 1/10

3s - loss: 0.0679 - val\_loss: 5.6680e-04

Epoch 2/10

2s - loss: 0.0043 - val\_loss: 0.0051

Epoch 3/10

2s - loss: 0.0037 - val\_loss: 0.0013

Epoch 4/10

2s - loss: 0.0030 - val\_loss: 5.0421e-04

Epoch 5/10

2s - loss: 0.0024 - val\_loss: 3.2711e-04

Epoch 6/10

2s - loss: 0.0025 - val\_loss: 2.8008e-04

Epoch 7/10

2s - loss: 0.0024 - val\_loss: 1.8258e-04

Epoch 8/10

2s - loss: 0.0026 - val\_loss: 3.4305e-04

Epoch 9/10

2s - loss: 0.0024 - val\_loss: 0.0012

Epoch 10/10

2s - loss: 0.0025 - val\_loss: 5.5239e-04

Train Score: 2.62 RMSE

Test Score: 7.64 RMSE

Training Time: 36.058888

Final Cross-Validation result: 0.000552388210277

Results

Delete result buffer