

 README.md

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 a note: 'All fields marked with (*) are required'. Below this, there are several input fields and buttons, each with a red arrow pointing to it from a yellow text box on the right. The fields and their corresponding annotations are:

- Load input filename (*) :** A text field with a 'Browse' button. Annotation: 'Browse to load input filename'.
- Save output filename as (*) :** A text field with a 'Browse' button. Annotation: 'Specify output filename'.
- Use config file :** A text field with a 'Browse' button. Annotation: 'upload config file'.
- Delete Config :** A button. Annotation: 'Delete uploaded config file'.
- Training percentage :** A text field with '0.7'. Annotation: 'Percentage of input data used for training'.
- Number of layers :** A text field with '3'. Annotation: 'Number of layers in the neural network'.
- Epoch :** A text field with '100'. Annotation: 'Number of passes through the whole training data'.
- Fraction of R_Units to be dropped [0.0, 1.0] :** A text field with '0.0'.
- Fraction of input units to be dropped [0.0, 1.0] :** A text field with '0.0'.
- Optimizer type :** A dropdown menu with 'adam' selected. Annotation: 'Select an optimizer from the menu -- {adam, SGD, RMSprop, nadam}'.
- Error Metric :** A dropdown menu with 'MSE' selected. Annotation: 'Choose a metric to measure error of the objective function -- {MSE}'.
- Layer dimensions [LAYERDIM1,LAYERDIM2,...] :** A text field with '1,4,1'. Annotation: 'Specify the no. of computation units per layer'.
- Learning rate (*) :** A text field with '0.0'. Annotation: 'Number that controls the step size of the weight update'.
- Momentum :** A text field with '0.0'. Annotation: 'Weighted sum of past gradients'.
- Append run configuration to logfile :** A checkbox. Annotation: 'Save the config as JSON which can loaded in future'.
- Save config :** A button with 'Save Config' text. Annotation: 'Save the config as JSON which can loaded in future'.
- Run :** A button. Annotation: 'Now build, train and evaluate model !'.

At the bottom of the GUI, there is a 'Delete Result Buffer' button and a footer that reads 'Developed by Abhishek Jain, Pradyumna Kaushik and Sreedhar Kumar'.

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