Documentation

All models are implemented using Python 2.7.3 and TensorFlow 1.3.

Usage

The following line will train a DRNN model and save it to the '/graphs' directory. See figure 1 for an overview of the architecture created. This line will call either 'train_DRNN' or 'train_LSTM'.

```
python main.py --bs 5 --hs 60 --nc 2 --lr 0.01 --ps 300 --ep
10000 --fs 150 --na 1,2,3 --ol 3 --dr True --td
train_file.tfrecord -- vd test_file.tfrecord
```

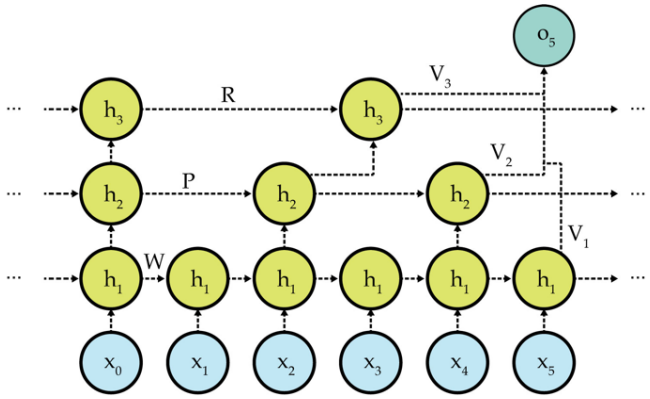

Figure 1: Overview of DRNN. x_i represents the input at time i, h_j represents the hidden state of recurrent unit j, and o_i represents the output at time i.

The model expects a '*.tfrecord' file for the train and validation datasets. Each sample of the dataset must be parsable by the function '_parse_function' defined in 'dataset_generator.py'.

--bs: Number of samples to consider in each gradient update

--hs: Size of the recurrent state

--nc: Number of different classes in the dataset file

--ps: Size to pad input to

--lr: Learning rate

--ep: Number of times the network will iterate the entire dataset

--fs: Size of the input vector

--na: A list with the delays of the recurrent units. [1,2,3] will create a network with 3 recurrent units, each with 1, 2 and 3 instants of delay respectively. Figure 1 represents this example.

--ol: Number of output layers. This increases depth

--dr: If True: trains DRNN else: trains LSTM

--td: Directory of the training dataset file.

--vd: Directory of the validation dataset file.

Implementation DRNN_model.py

```
train_DRNN(batch_size = 5,
                hidden_size = 130,
                num_classes = 2,
                learning_rate = 0.01,
                padding_size = 300,
                num_epochs = 10000,
                feature_size = 150,
                net_arch = [1,4,8,12],
                out_layers = 3)
```

Defined in https://github.com/DeadlyBunny24/RNN-ActivityRecognition

Args:

- batch_size: Integer value. Number of samples to consider for one gradient update.
- hidden_size: Integer value. Size of the hidden layer.
- num_classes: Integer value. Number of distinct classes in the dataset.
- start_learning_rate: Floating point value. Starting learning rate. The network will slowly
- learning_rate: Floating point value. Learning rate of the optimizer.
- padding_size: Integer value. Size to pad the sequence to.
- num_epochs: Integer value. Number times the model will iterate the entire dataset.
- feature_size: Integer value. Size of the input vector.
- net_arch: Integer value list. Set of recurrent unit delays. Each element will instantiate a new recurrent unit that will trigger every list[i] instances. See figure 1 for an illustration.
- out_layers: Integer value. Number of hidden layers.

Returns:

None. The trained model is saved to the graphs folder as: ‘DRNN_na_{num_samples}_lr_{learning_rate}_hs_{hidden_size}_bs_{batch_size}_ol_{out_layers}’

Implementation LSTM_model.py

```
train_LSTM(batch_size = 5,
                hidden_size = 130,
                num_classes = 2,
                learning_rate = 0.01,
                padding_size = 300,
                num_epochs = 10000,
                feature_size = 150,
                net_arch = [1,4,8,12],
                out_layers = 3)
```

Defined in https://github.com/DeadlyBunny24/RNN-ActivityRecognition

Args:

- batch_size: Integer value. Number of samples to consider for one gradient update.
- hidden_size: Integer value. Size of the hidden layer.
- num_classes: Integer value. Number of distinct classes in the dataset.
- start_learning_rate: Floating point value. Starting learning rate. The network will slowly
- learning_rate: Floating point value. Learning rate of the optimizer.
- padding_size: Integer value. Size to pad the sequence to.
- num_epochs: Integer value. Number times the model will iterate the entire dataset.
- feature_size: Integer value. Size of the input vector.
- net_arch: Integer value list. List of the recurrent unit delays. Each element will instantiate a new recurrent unit that will trigger every list[i] instances. See figure 1 for an illustration.
- out_layers: Integer value. Number of hidden layers.

Returns:

None. The trained model is saved to the graphs folder as: 'LSTM_na_{num_samples}_lr_{learning_rate}_hs_{hidden_size}_bs_{batch_size}_ol_{out_layers}'