

Artificial Neural Networks

Running Instructions

There are two trained models, FCNN(`fcnn.model`) and CNN(`fer2013_ResNet20v1_model.89-0.66.h5`), in the directory `models/` . The interfaces designed to load models and classify images can be found in the `src/test.py` :

1. **FCNN:** `test_fer_model(img_folder, model_path)`
2. **CNN:** `test_deep_fer_model(img_folder, model_path)`

The first parameter of this function is the path to the testing images, and the second is the path to the trained model. This function will return a vector of predictions of shape (N,) with N being the number of the test samples. All test samples will be read lexicographically. **Especially**, the CNN model is trained using `keras` so please make sure the testing environment has `keras` installed.

One thing to note is that all test samples will be subtracted by the mean of training samples, which is pre-calculated and stored in the `datasets/fer2013_npy/train_samples_mean.npy` , so please make sure this file is used, otherwise the prediction will be significantly affected. If you change the path of `train_samples_mean.npy` , please also change the global variable `train_samples_mean_path` in `src/test.py` correspondingly.

Dependencies

To ensure the program work, the libraries below should be installed in the running environment:

1. `keras==2.1.4`
2. `h5py==2.7.1`
3. `numpy==1.13.3`
4. `matplotlib==2.1.1`
5. `cycler==0.10.0`
6. `pyparsing==2.2.0`
7. `python-dateutil==2.6.1`
8. `pytz==2017.3`
9. `six==1.11.0`
10. `future==0.16.0`

Simply, all environment requirements can be installed by `pip install requirements` .

Fine-tune Hyper-parameters

We implement a script, `src/utils/fcnn_optimizer.py`, to semi-automatically search the best hyper-parameters using stage optimization strategy. Basically, you can leave it the jobs of searching parameters within the given range but still need to update the range, number of iterations and number of epochs for each stage manually. The constant variables are reserved as the shared configurations for each iteration.

To run the script, you should firstly turn on, i.e. uncomment or add, the corresponding function calling and turn off, i.e. comment or remove, the remaining function callings. The results will be recored and stored in the directory `params/`.

Dataset

The all involved datasets are stored in the directory `datasets/` and loaded by the module `src/utils/data_utils.py`. In order to accelerate the loading speed, one of the existing datasets, *FER2013*, is also stored as a form of `npz` supported by the library `numpy`.

If you want to load your **custom datasets**, the only change concerning the data loading is updating the directory path in the `src/utils/data_utils.py`.

Results of Experiments

The raw results of experiments are all recorded and stored in the separated directories:

1. **evaluations**: storing the evaluation results, confusion matrix, f1 measure and classification rate etc., generated by the module `src/evaluator.py`
2. **checkpoints**: storing the saved checkpoints of hyper-parameters during the training if turning on the checkpoint mode.
3. **models**: storing the saved binary files of trained models
4. **figures**: storing the figures, like loss_accuracy figure, generated by the module `src/evaluator.py`
5. **optimizations**: the visualization of results of each stage for stage optimization
6. **params**: storing the text results of stage optimization and the best value for particular stage.

The detailed analysis for all experiments can be found in the report file `nn_report.pdf`.