

Remoore Sensing - Keeping track of experiments

Problem Overview

We want to be able to replay any experiment made and to review and understand an experiment without actually having to replay it. This to keep scientific transparency while keeping in mind any experiment made.

We have the following constraints :

1. Json storage. It has been previously decided to store the data in a json format.
2. Each experiment note should contain all the information about an experiment, it should contain a free form field. Experiment notes are not fixed form but most contain about the same information.
3. The number of experience will probably be more than 10 less than 100,000. This poses no storage problem as an experiment can be described with less that 10,000 characters (less that 1Go for 100k experiments). But requires some automation
4. Integrity of data (The record of experiments cannot be lost as it would reduce tremendously the scientific value of any previous work.)
5. Should be easy. Manipulating meta-data is not fun but it should be done so the task of adding/reviewing experiments must be as simple as possible.

Solution Architecture

The experiment track will be stored in a single json file with the following structure :

```
{
  "experiments":{
    "title1":{},
    "title2":{}
  }
}
```

The experiments `title1` & `title2` may contain any information. They should in particular contain information to review and replay experiments. An utility software will guide the user through adding an experiment to the database. The database will be stored in a trustful cloud to ensure it's security. This simple architecture fulfills all discussed constraints.

Solution Implementation

The implementation I propose is composed of two files (a python script and a `database.json`) in a git repo and a template file.

The user calls the python script in the repo passing a path to a new experiment note (a single json file that describe one experience and that can be build using the template file). Then the script adds the experience to the database and saves the database on github.

Tutorial

After doing an hypothetical experiment I fill the template file.

```

1 {
2   "title": "Experiment made for the tutorial",
3   "date": "8 jan 2022",
4   "author": "A Martin",
5   "notes": "This is not an experiment, it's a test of the sync between my poc and git hub.",
6   "architecture": {
7     "name": "None",
8     "dropout": false,
9     "regularization": false,
10    "batch_normalization": false,
11    "data_preprocessing": [],
12    "data_agumentation": []
13  },
14  "optimization": {
15    "algorithm": "",
16    "hyperparameters": {
17    }
18  },
19  "training": {
20    "time": 0,
21    "epochs": 0,
22    "train_val_ratio": 0
23  },
24  "evaluation": {
25    "metric": "",
26    "results": {
27      "loss": 0.0,
28      "accuracy": 0.0,
29      "precision": 0.0,
30      "recall": 0.0,
31      "f1_score": 0.0,
32      "auc_roc": 0.0
33    }
34  },
35  "reproducing_experiment": {
36    "dataset_version": "",
37    "dataset": "",
38    "github_link": ""
39  },
40  "hardware_specifications": {
41    "operating_system": "ubuntu",
42    "processor": "?",
43    "memory": "?",
44    "libraries_and_frameworks": [
45      "keras",
46      "tensorflow"
47    ]
48  }

```

Then I call the script passing as an argument new_experiment.json (the pass to the above file)

```

(base) arno@Momo-linux:~/Téléchargements/PSA January work/test command/Write app
$ python3 ./easy_experiment_backup_poc/add_experiments.py new_experiment.json

```

Then looking on github we can see the experiment has been added to the data base.

```

{"experiments": [{"Not an experiment": {"title": "Not an experiment", "date": "8 jan 2022", "author": "A Martin", "notes": "This is a Test not an experiment", "architecture": {"name": "None", "dropout": false, "regularization": false, "batch_normalization": false, "data_preprocessing": [], "data_agumentation": []}, "optimization": {"algorithm": "", "hyperparameters": {}}, "training": {"time": 0, "epochs": 0, "train_val_ratio": 0}, "evaluation": {"metric": "", "results": {"loss": 0.0, "accuracy": 0.0, "precision": 0.0, "recall": 0.0, "f1_score": 0.0, "auc_roc": 0.0}}, "reproducing_experiment": {"dataset_version": "", "dataset": "", "github_link": ""}, "hardware_specifications": {"operating_system": "ubuntu", "processor": "?", "memory": "?", "libraries_and_frameworks": ["keras", "tensorflow"]}}, {"Test Github sync": {"title": "Test Github sync", "date": "8 jan 2022", "author": "A Martin", "notes": "This is not an experiment, it's a test of the sync between my poc and git hub.", "architecture": {"name": "None", "dropout": false, "regularization": false, "batch_normalization": false, "data_preprocessing": [], "data_agumentation": []}, "optimization": {"algorithm": "", "hyperparameters": {}}, "training": {"time": 0, "epochs": 0, "train_val_ratio": 0}, "evaluation": {"metric": "", "results": {"loss": 0.0, "accuracy": 0.0, "precision": 0.0, "recall": 0.0, "f1_score": 0.0, "auc_roc": 0.0}}, "reproducing_experiment": {"dataset_version": "", "dataset": "", "github_link": ""}, "hardware_specifications": {"operating_system": "ubuntu", "processor": "?", "memory": "?", "libraries_and_frameworks": ["keras", "tensorflow"]}}, {"Experiment made for the tutorial": {"title": "Experiment made for the tutorial", "date": "8 jan 2022", "author": "A Martin", "notes": "This is not an experiment, it's a test of the sync between my poc and git hub.", "architecture": {"name": "None", "dropout": false, "regularization": false, "batch_normalization": false, "data_preprocessing": [], "data_agumentation": []}, "optimization": {"algorithm": "", "hyperparameters": {}}, "training": {"time": 0, "epochs": 0, "train_val_ratio": 0}, "evaluation": {"metric": "", "results": {"loss": 0.0, "accuracy": 0.0, "precision": 0.0, "recall": 0.0, "f1_score": 0.0, "auc_roc": 0.0}}, "reproducing_experiment": {"dataset_version": "", "dataset": "", "github_link": ""}, "hardware_specifications": {"operating_system": "ubuntu", "processor": "?", "memory": "?", "libraries_and_frameworks": ["keras", "tensorflow"]}}}]}

```

