Welcome to the Thales AI Demonstrator !

Getting Started

First of all, these instructions will get you a copy of the project up and running on your local machine.

Prerequisites

This projects runs with python. If you haven't got python on your machine, install it ! (https://www.python.org/downloads/).

> Note : On Windows, you will need python3.5 64 bits. Else, you will not be able to install `tensorflow`. Here's a link of a working version : https://www.python.org/downloads/release/python-354/. On Mac you can work with Python 3.5 or 3.6, even though 3.5 is recommended.

This project also uses WebSockets managed by the Django Framework. In order for those to work, you will need a `Redis` server instance running on your computer.

Windows	Mac (Or other UNIX OS)
Using `Redis` on Windows is quite tricky, so you have to use a specific release for it to work with Django Channels. If you don't have it, download the archive from GitHub, it is available HERE. Then, extract the `Redis-x64-3.0.504.zip` archive in another folder located wherever you want. (In `C:\Program Files\Redis` for example). Once that is done, you can either add the folder path to the PATH environment variable (instructions below). If you do not want to add it to your path, you can modify the start file in any text editor, and add the path to `Redis` folder to the last line. (Change from `redis-server` to `C:\"Program Files"\Redis\redis-server` for example)	Simply download Redis from here (https://redis.io/). Detailed installation instructions can be found at https://redis.io/download/#installation.

This projects client side runs using `NodeJS` and `NPM`. If you haven't got those on your machine, install both of them !

- Instructions to install both : https://www.npmjs.com/get-npm

Installing the project

The first thing you will have to do, even before installing the project, is decide where you will want to save the data you collect and the models you compile. It can be anywhere you choose, you just have to specify the path to these folders in the `config.json` file, at the very root of the project. If you have any trouble with that, refer to the dedicated section below.

> *Windows* : You will need a wheel for the Twisted python package. If it does not come with the project, you can find it at https://www.lfd.uci.edu/~gohlke/pythonlibs/#twisted. Download the `Twisted-18.9.0-cp35-cp35m-win_amd64.whl` file (for 64 bits, for 32 bits it's the `Twisted-18.9.0-cp35-cp35m-win_32.whl` file) and add it at the very root of the project.

> *Unix* : Before moving on, you will have to change the path in all bash scripts to your project path. Just update the first line of each script from `unix_scripts` folder.

Once you've done that, execute the `install` script corresponding to your OS (located in unix_scripts for unix systems, or win_scripts for Windows). You just need to double-click the script file to do that.

> *Note* : If this is the first time you run the `install` script, make sure you have a working internet connection, as the script will have to download some packages. This might take a few minutes.

> *Unix* : On unix systems, you will have to edit all scripts to add the path to the project folder on the third line. Just replace `cd /path/to/project` by the path of your project. For example it could be `cd /Users/username/Documents/Thales/thales-project`.

Start the demonstrator

Now that you've installed everything, you're ready to go ! To start the demonstrator, you have to run the `start` script corresponding to your OS (located in unix_scripts for unix systems, or win_scripts for Windows). Then, to get to the app, just open a browser on http://localhost:4200/.

To know more about the possibilities with the demonstrator, and how to use it, check the USER MANUAL out !

Additional instructions

Add Redis to PATH

To access it, press `WIN+R` and execute `sysdm.cpl`. Then, in Advanced System Properties check environment variables. Finally, in PATH variable, add a new path, being the path to redis. For example, add `C:\Program Files\Redis` to path if `redis-server.exe` is in this Redis folder

Writing your config file

Making this config file is easy. You just have to copy the `config.example.json`, rename it to `config.json`, and then fill out the paths from the `PATH` section with your preferred paths.

When writing the path to a folder, note that the folder must exist. Create it if it doesn't exist !

> *Windows* : On Windows, path will look like this : `"C:\\Program Files\\Thales\\Demo\\data"`

> *Unix* : On unix systems (like mac or linux), path will look like this : `"/Users/username/Thales/Demo/data"`

Here's an example of a filled out config file on OSX :

```
{
    "PATH": {
        "data": "/Users/username/Documents/Thales/thales-project/data",
        "logs": "/Users/username/Documents/Thales/thales-project/logs",
        "pkl": "/Users/username/Documents/Thales/thales-project/data/pkl",
        "pkl2": "/Users/username/Documents/Thales/thales-
project/data/pkl2",
        "weights": "/Users/username/Documents/Thales/thales-
project/data/weights"
    },
    "VARS": {
        "cycles_per_batch": 5,
        "cycles_per_batch_mix_method": 100,
        "time_step_ms": 500
    },
    "URLS": {
        "BASE": {
            "back": "http://localhost:",
            "front": "http://localhost:"
        },
        "PORT": {
            "back": 8000,
            "front": 4200
        }
    }
}
```

If you later want to modify your config file, to change the `cycles_per_batch` parameter for example, don't forget to run the `install` script once again.

> *Note* : This will copy the config file to the `back_django`, `front` and `ml_tools/src` folders, to make sure it is installed properly in every package. You do not need an internet connection to perform this.