

Step1: Fill up screening question

Step2: Fill up the consent form

Step3: Download study materials

Step 4: Finish Session #1 and Session #2 user study

Step 5: Fill up the survey

Session #1:

- This session consists of a user study using Session1_1 and Session 1_2 folders from the downloaded ContractStudy.zip file.
- In each folder, execute 3 buggy programs (BuggyProgram1.py, BuggyProgram2.py, BuggyProgram3.py) according to the given instructions in each folder.
- Set #1 (Session1_1) buggy programs will be executed using regular Keras and execute Set #2 (Session1_2) buggy programs will be executed using Keras with annotated DL contracts, we named it @Keras.

Task #1:

- Execute BuggyProgram1.py, BuggyProgram2.py, BuggyProgram3.py from both sessions directories (Session1_1 and Session 1_2).
- Detect the bugs in the buggy programs by following specific instructions in Session1_2. In Session1_1, you can observe the output without any error message and should find the accuracy is low or training time is high. Because there is a silent bug in each of the codes which regular Keras does not warn us but annotated @Keras does.
- In Session1_2, you can observe the error message after "contracts.interface.ContractException:". Fix those 3 buggy programs according to the provided suggestion only in Session1_2 and execute after fix.
- Fill up a survey after executing Session1_1 and Session 1_2 buggy programs.

Session #2:

- Follow a tutorial with an example on how to write contracts on top of a Keras API is provided below.
- Then, write 3 contracts using provided instructions in Session 2 (Task 2.1, Task 2.2, Task 2.3).

Task #2:

- Write contracts using Deep Learning contract syntax following the provided guidelines (Task 2.1, Task 2.2).
 - Task 2.1: Write a contract for multiclass classification on top of compile API (line 301 in the training.py) by defining activation_func_multiclass as 'softmax', 'categorical_crossentropy' as a loss function.
 - Task 2.2: Write a contract for data normalization on top of fit API (~line 670 in the training.py) by defining 'data_normalized' context as string, 'normalization_interval' less than or equal to 2.0 as float.

- Fill up a survey on the process and experience of writing Deep Learning contracts.

After completing the two sessions, please delete all the files of this user study.

Tutorial:

The following tutorial will guide you through how to write a contract on top of compile API (line 301 in the training.py file inside AugKeras folder).

We need to first define last_layer and activation_func_binary and binary_crossentropy string as below:

```
last_layer = 'last_layer'  
activation_func_binary = 'sigmoid'  
binary_crossentropy = "binary_crossentropy"
```

Then we annotate binary classification contract as below on top of compile API:

```
@contract(context='str,=$last_layer',  
activation='str,=$activation_func_binary',  
loss_func='str,=$binary_crossentropy')
```

Here, context is specified as 'last_layer', 'activation' has been specified with 'activation_func_binary' and 'loss_func' has been specified as loss_func with the values declared above.