

Transpile

Transpilation is a crucial process in the deployment of Verifiable Machine Learning models and its powered by is powered by [ezkl](#) . It involves the usage of an ONNX model and input data to perform the setup . With this setup we can generate proofs that can be verified, ensuring the integrity and reliability of the model's predictions.

The transpilation process begins by reading the model and input data from the specified path. The model and data are then sent for transpilation. By default, the output of this process is managed by Giza for later use.

...

Copy

```
giza transpile --framework EZKL --input-data input.json awesome_model.onnx [giza][2023-12-04 19:40:11.274]
No model id provided, checking if model exists ✓ [giza][2023-12-04 19:40:11.275] Model name is:
awesome_model [giza][2023-12-04 19:40:11.424] Model Created with id -> 1! ✓ [giza][2023-12-04 19:40:11.944]
Sending model for setup ✓ [giza][2023-12-04 19:40:13.046] Setup job created with name 'ezkl-job-20231204-
32a4a1c1' and id -> 1 ✓ [giza][2023-12-04 19:41:54.277] Setup job is successful ✓
```

...

The result of the transpilation process is the following:

- Circuit settings
- Proving key
- Verification key
-

How do we transpile a model

There are three main methods for transpiling a model:

Method 1: Using the giza transpile --framework EZKL command

- This is the simplest method and is recommended for most users.
- When you run this command, Giza handles everything for you.
- It first checks if a model with the specified name already exists. If not, it creates a new model and then transpiles it.
- The output of this process is managed by Giza.
- This is the strategy that we followed in the example before.
-

Method 2: Manually creating a model and then transpiling it

- This method gives you more control over the process.
- First, you create a model manually using the `giza models create` command.
- After the model is created, you can transpile it using the `giza transpile --framework EZKL --model-id ...` command.
- This method is useful when you want to specify particular options or parameters during the model creation and transpilation process.
-

...

Copy

```
giza models create --name awesome_model --description "A Model for testing different models" [giza][2023-09-13
14:04:59.532] Creating model ✓ { "id": 2, "name": "awesome_model", "description": "A Model for testing different
models" }
```

...

...

Copy

```
giza transpile --framework EZKL --model-id 2 --input-data input.json awesome_model.onnx [giza][2023-12-04
19:40:10.646] Model found with id -> 2! ✓ [giza][2023-12-04 19:40:12.176] Sending model for Setup ✓ [giza]
[2023-12-04 19:40:13.046] Setup job created with name 'ezkl-job-20231204-32a4a1c1' and id -> 13 ✓ [giza]
```

[2023-12-04 19:41:54.277] Setup job is successful ✓

...

Method 3: Using a previous model

- If you have a previously created model, you can transpile it by indicating the model-id in thegiza transpile --framework EZKL --model-id ...
- orgiza versions transpile --framework EZKL --model-id
- command.
- This method is useful when you want to create a new version of an existing model.
- The output of the transpilation process is saved in the same location as the original model.
-

...

Copy

Using the previous model (id:2) we can transpile a new model, which will create version 2 of the model.

```
giza transpile --framework EZKL --model-id 2 --input-data input.json awesome_model.onnx [giza][2023-12-04 19:40:10.646] Model found with id -> 2! ✓ [giza][2023-12-04 19:40:12.176] Sending model for Setup ✓ [giza][2023-12-04 19:40:13.046] Setup job created with name 'ezkl-job-20231204-123fadc1' and id -> 14 ✓ [giza][2023-12-04 19:41:54.277] Setup job is successful ✓
```

...

What is happening with the models and versions

In Giza, a model is essentially a container for versions. Each version represents a transpilation of a machine learning model at a specific point in time. This allows you to keep track of different versions of your model as it evolves and improves over time.

To check the current models and versions that have been created, you can use the following steps:

1. Use thegiza models list
2. command to list all the models that have been created.
3. For each model, you can use thegiza versions list --model-id ...
4. command to list all the versions of that model.
- 5.

Remember, each version represents a specific transpilation of the model. So, if you have made changes to your machine learning model and transpiled it again, it will create a new version.

This system of models and versions allows you to manage and keep track of the evolution of your machine learning models over time.

For example, let's say you have created a model called awesome_model and transpiled it twice. This will create two versions of the model, version 1 and version 2. You can check the status of these versions using thegiza versions list --model-id ... command.

...

```
Copy giza versions list --model-id 29 [giza][2023-12-04 14:17:08.006] Listing versions for the model ✓ [ { "version": 1, "size": 52735, "status": "COMPLETED", "message": "Setup Successful!", "description": "Initial version", "created_date": "2023-09-13T12:08:38.177605", "last_update": "2023-09-13T12:08:43.986137" }, { "version": 2, "size": 52735, "status": "Setup", "message": "Transpilation Successful!", "description": "Initial version", "created_date": "2023-12-04T12:11:30.165440", "last_update": "2023-12-04T12:11:31.625834" } ]
```

...

[Previous EZKL](#) [Next Deploy](#)

Last updated 1 month ago