

IDevice

Package: android.hardware.neuralnetworks@1.0

interface IDevice

This interface represents a device driver.

Methods

getCapabilities

getCapabilities ()
generates (ErrorStatus status, Capabilities capabilities)



Gets the capabilities of a driver.

Details

Generates	status	Error status of the call, must be:- NONE if successful - DEVICE_UNAVAILABLE if driver is offline or busy - GENERAL_FAILURE if there is an unspecified error
	capabilities	Capabilities of the driver.

getSupportedOperations

getSupportedOperations (Model model)
generates (ErrorStatus status, vec<bool> supportedOperations)



Gets the supported operations in a model.

getSupportedSubgraph indicates which operations of a model are fully supported by the vendor driver.If an operation may not be supported for any reason, getSupportedOperations must return false for that operation.

Details

Parameters	model	A model whose operations--and their corresponding operands--are to be verified by the driver.
Generates	status	Error status of the call, must be: NONE if successful - DEVICE_UNAVAILABLE if driver is offline or busy - GENERAL_FAILURE if there is an unspecified error - INVALID_ARGUMENT if provided model is invalid
	supportedOperations	A list of supported operations, where true indicates the operation is supported and false indicates the operation is not supported. The index of "supported" corresponds with the index of the operation it is describing.

prepareModel

```
prepareModel (Model model, IPreparedModelCallback callback)
generates (ErrorStatus status)
```



Creates a prepared model for execution.

prepareModel is used to make any necessary transformations or alternative representations to a model for execution, possibly including transformations on the constant data, optimization on the model's graph, or compilation into the device's native binary format. The model itself is not changed.

The model is prepared asynchronously with respect to the caller. The prepareModel function must verify the inputs to the prepareModel function are correct. If there is an error, prepareModel must immediately invoke the callback with the appropriate ErrorStatus value and nullptr for the IPreparedModel, then return with the same ErrorStatus. If the inputs to the prepareModel function are valid and there is no error, prepareModel must launch an asynchronous task to prepare the model in the background, and immediately return from prepareModel with ErrorStatus::NONE. If the asynchronous task fails to launch, prepareModel must immediately invoke the callback with ErrorStatus::GENERAL_FAILURE and nullptr for the IPreparedModel, then return with ErrorStatus::GENERAL_FAILURE.

When the asynchronous task has finished preparing the model, it must immediately invoke the callback function provided as an input to prepareModel.If the model was prepared successfully, the callback object must be invoked with an error status of `ErrorStatus::NONE` and the produced `IPreparedModel` object.If an error occurred preparing the model, the callback object must be invoked with the appropriate `ErrorStatus` value and nullptr for the `IPreparedModel`.

The only information that may be unknown to the model at this stage is the shape of the tensors, which may only be known at execution time.As such, some driver services may return partially prepared models, where the prepared model can only be finished when it is paired with a set of inputs to the model.Note that the same prepared model object can be used with different shapes of inputs on different(possibly concurrent)executions.

Multiple threads can call prepareModel on the same model concurrently.

Details

Parameters	model	The model to be prepared for execution.
	callback	A callback object used to return the error status of preparing the model for execution and the prepared model if successful, nullptr otherwise.The callback object's notify function must be called exactly once, even if the model could not be prepared.
Generates	status	Error status of launching a task which prepares the model in the background;must be:- NONE if preparation task is successfully launched - <code>DEVICE_UNAVAILABLE</code> if driver is offline or busy - <code>GENERAL_FAILURE</code> if there is an unspecified error - <code>INVALID_ARGUMENT</code> if one of the input arguments is invalid

getStatus

getStatus ()



generates (DeviceStatus status)

Returns the current status of a driver.

Details

Generates	status	Status of the driver, one of:- DeviceStatus::AVAILABLE - DeviceStatus::BUSY - DeviceStatus::OFFLINE - DeviceStatus::UNKNOWN
-----------	---------------	---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

上次更新日期：十二月 8, 2017