Type Method

makeContext(options:_:)

Returns a new context that wraps a graph object that the given closure defines.

iOS 26.0+ | iPadOS 26.0+ | Mac Catalyst | macOS 26.0+ | tvOS 26.0+ | visionOS 26.0+ | watchOS 26.0+

```swift
static func makeContext(
    options: BNNSGraph.CompileOptions = CompileOptions(),
    _ block: (inout BNNSGraph.Builder) -> [any BNNSGraph.TensorDescriptor]
) throws -> BNNSGraph.Context
```

Parameters

options

The compilation options.

block

A closure with a `BNNSGraph.Builder` parameter that points to the BNNS Graph builder.

Discussion

Use this function to specify the operations that define a BNNS graph. For example, the following code creates a graph that performs element-wise multiplication of two eight-element vectors:

```swift
let context = try BNNSGraph.makeContext ({
    builder in

    let x = builder.argument(name: "x",
                             dataType: Float.self,
                             shape: [8])
```

```
        let y = builder.argument(name: "y",
                                 dataType: Float.self,
                                 shape: [8])


        let z = x * y


        return [z]
    })
```

The following code defines the arguments and executes the graph:

```
    var args = context.argumentNames().map {
        name in


        return context.tensor(argument: name,
                              fillKnownDynamicShapes: false)!
    }


    defer {
        args.forEach {
            $0.deallocate()
        }
    }


    args[context.argumentPosition(argument: "x")]
        .allocate(initializingFrom: [1, 2, 3, 4, 5, 6, 7, 8] as [Float])


    args[context.argumentPosition(argument: "y")]
        .allocate(initializingFrom: [8, 7, 6, 5, 4, 3, 2, 1] as [Float])


    // Output argument
    args[0].allocate(as: Float.self, count: 8)


    try context.executeFunction(arguments: &args)
```

On return, args[0] contains the values [8.0, 14.0, 18.0, 20.0, 20.0, 18.0, 14.0, 8.0].

See Also

Building graphs in Swift

`struct Builder`

> A structure thats provides a closure you can use to define the arguments and operations of a BNNS Graph.

`struct Tensor`

> A structure that represents an abstract handle to a tensor that you use within a `BNNSGraph.makeContext` closure.

`{}` Supporting real-time ML inference on the CPU

> Add real-time digital signal processing to apps like Logic Pro X and GarageBand with the BNNS Graph API.