Structure

BNNSGraph.Builder

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

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

```
struct Builder
```

Topics

Protocols

`protocol OperationParameter`

A protocol that allows functions to accept either tensors or collections.

`protocol SliceIndex`

A protocol that the BNNS graph builder uses to specify slice indices.

Structures

`struct SliceRange`

A structure that represents a range.

`struct Tensor`

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

Instance Methods

```
func argument<T>(name: String?, dataType: T.Type, shape: [Int], intent:
BNNSGraph.Builder.Intent) -> BNNSGraph.Builder.Tensor<T>
```

Registers and returns an input or in-out tensor argument to the graph.

```
func concatenate<T>([BNNSGraph.Builder.Tensor<T>], axis: Int) ->
BNNSGraph.Builder.Tensor<T>
```

Adds a concatenation operation to the current graph.

```
func constant<T>(name: String?, value: T) -> BNNSGraph.Builder.Tensor<T
>
```

Registers and returns a tensor that contains a constant scalar value.

```
func constant<T>(name: String?, values: some AccelerateBuffer, shape: [
Int]?) -> BNNSGraph.Builder.Tensor<T>
```

Registers and returns a tensor with the specified shape that contains constant data, such as weight or bias values.

```
func constant(values: Array<Array<Float>>, rowMajor: Bool) -> BNNSGraph
.Builder.Tensor<Float>
```

Returns a rank 2 tensor from an array of arrays.

```
func constant(values: Array<Array<Float16>>, rowMajor: Bool) ->
BNNSGraph.Builder.Tensor<Float16>
```

Type Aliases

```
typealias PoolingPadding
```

The padding that you use for pooling operations to specify zero-padding.

Enumerations

```
enum Activation
```

The activation function that a recurrent operation uses.

```
enum CeilingMode
```

The pooling ceiling mode.

```
enum ConvolutionPadding
```

The padding that you use for convolution operations to specify zero-padding.

`enum Direction`

The direction of a recurrent operation.

`enum Intent`

Constants that describe argument intents.

`enum Padding`

The padding that you use for pad operations.

`enum PoolingFunction`

The pooling function

`enum ScatterMode`

Constants that specify how scatter operations overwrite destination elements.

`enum SortOrder`

The sort order for functions such as `argsort`.

See Also

Building graphs in Swift

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

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

`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.