

[Accelerate](#) / BNNSRandomFillNormalFloat(\_:\_:\_:\_:)

## Function

# BNNSRandomFillNormalFloat(\_:\_:\_:\_:)

Fills the specified tensor with random floating-point values mapped to a normal distribution.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | tvOS 16.0+ | visionOS 1.0+ | watchOS 9.0+

```
func BNNSRandomFillNormalFloat(  
    _ generator: BNNSRandomGenerator?,  
    _ desc: UnsafeMutablePointer<BNNSNDArrayDescriptor>,  
    _ mean: Float,  
    _ stddev: Float  
) -> Int32
```

## Parameters

### generator

The random number generator.

### desc

The descriptor of the destination.

### mean

The mean of the distribution.

### stddev

The standard deviation of the distribution.

## Discussion

Use this function to fill an array descriptor with random values mapped to a normal distribution.

If you use the same generator on multiple threads, note that this function serializes the generator through an internal lock. To eliminate this contention, use different generators for each thread.

The following code populates a descriptor with random floating-point values:

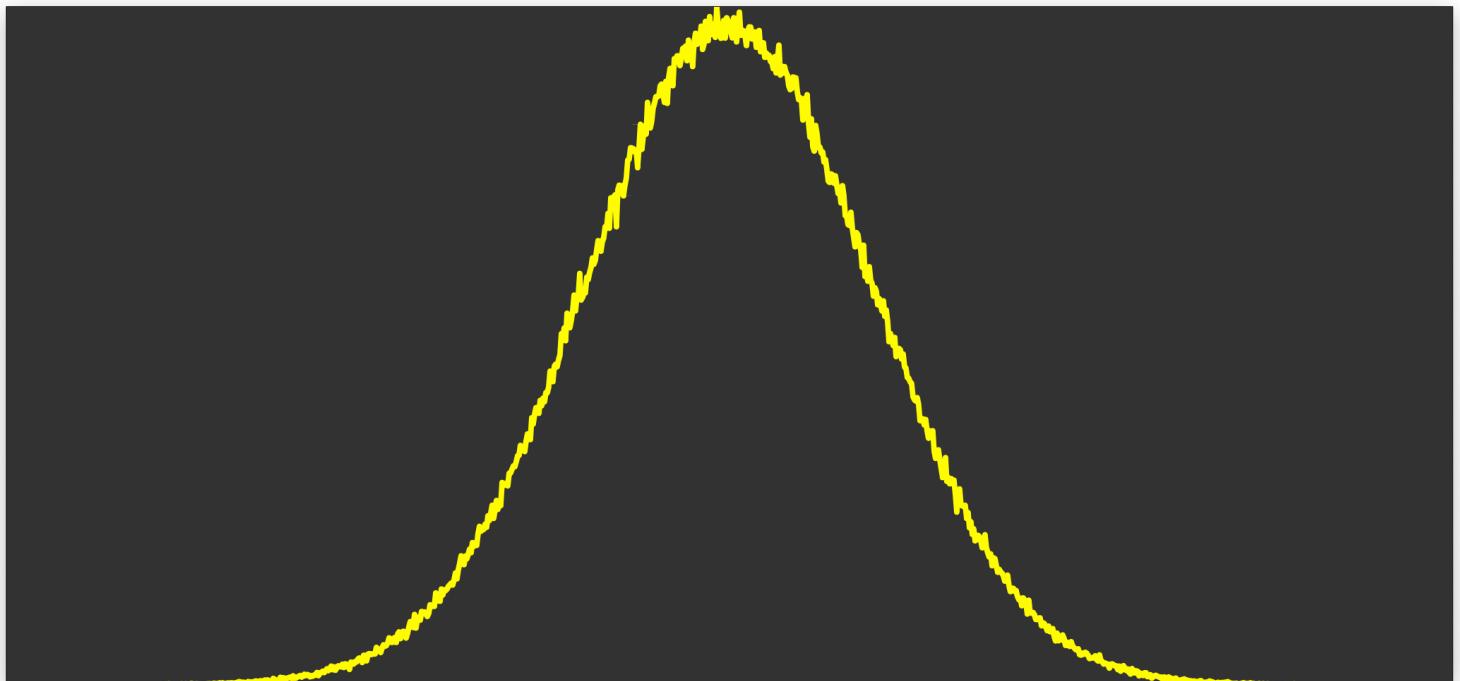
```
let width = 1024
let height = 1024

var descriptor = BNNSNDArrayDescriptor.allocateUninitialized(
    scalarType: Float.self,
    shape: .matrixRowMajor(width,
                           height))

let randomNumberGenerator = BNNSCreateRandomGenerator(
    BNNSRandomGeneratorMethodAES_CTR,
    nil)

BNNSRandomFillNormalFloat(randomNumberGenerator,
                           &descriptor,
                           0,
                           2)
```

The graph below shows the distribution of values in descriptor.



# See Also

## Random number generation

`class RandomGenerator`

A random number generator.

`func BNNSCreateRandomGenerator(BNNSRandomGeneratorMethod, UnsafePointer<BNNSFilterParameters>?) -> BNNSRandomGenerator?`

Returns a new random number generator using an internally generated random seed.

`func BNNSCreateRandomGeneratorWithSeed(BNNSRandomGeneratorMethod, UInt64, UnsafePointer<BNNSFilterParameters>?) -> BNNSRandomGenerator?`

Returns a new random number generator using the specified seed.

`struct BNNSRandomGeneratorMethod`

Constants that describe random number generation methods.

`typealias BNNSRandomGenerator`

A pointer to a random number generator object.

`func BNNSRandomFillUniformInt(BNNSRandomGenerator?, UnsafeMutablePointer<BNNSNDArrayDescriptor>, Int64, Int64) -> Int32`

Fills the specified tensor with random integer values from the continuous uniform distribution within a range.

`func BNNSRandomFillUniformFloat(BNNSRandomGenerator?, UnsafeMutablePointer<BNNSNDArrayDescriptor>, Float, Float) -> Int32`

Fills the specified tensor with random floating-point values from the continuous uniform distribution within a range.

`func BNNSRandomFillCategoricalFloat(BNNSRandomGenerator?, UnsafePointer<BNNSNDArrayDescriptor>, UnsafePointer<BNNSNDArrayDescriptor>, Bool) -> Int32`

Fills the specified tensor with random values from the categorical distributions with the given event probabilities.

`func BNNSRandomGeneratorStateSize(BNNSRandomGenerator?) -> Int`

Returns the state size, in bytes, of a random number generator.

```
func BNNSRandomGeneratorGetState(BNNSRandomGenerator?, Int, Unsafe  
MutableRawPointer) -> Int32
```

Returns the state of a random number generator.

```
func BNNSRandomGeneratorSetState(BNNSRandomGenerator?, Int, Unsafe  
MutableRawPointer) -> Int32
```

Sets the state of a random number generator.

```
func BNNSDestroyRandomGenerator(BNNSRandomGenerator?)
```

Destroys a random number generator.