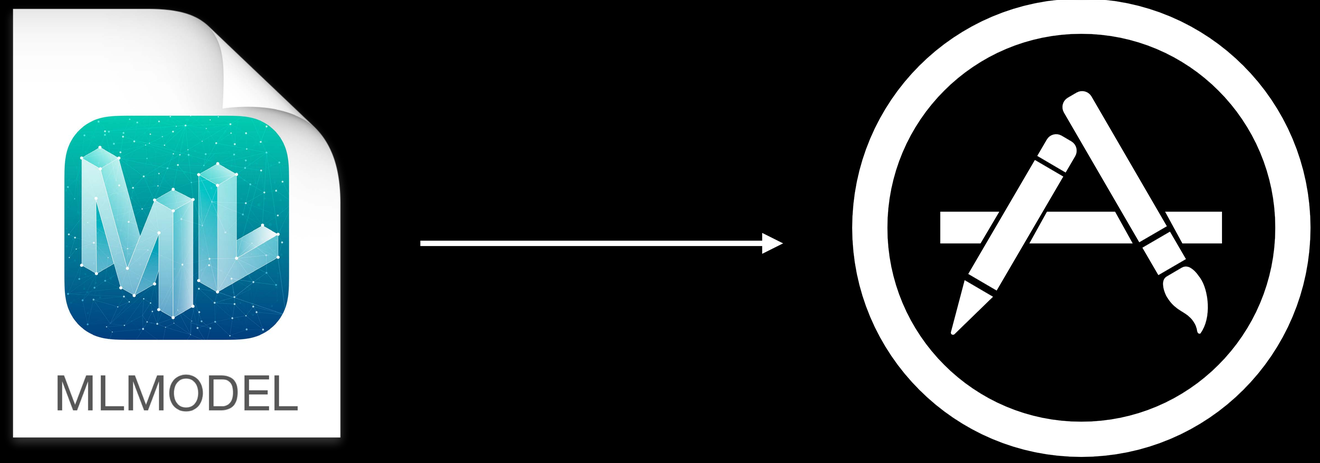
#WWDC18

What's New in Core ML

Part one

Michael Siracusa, Core ML
Francesco Rossi, Core ML
Bill March, Core ML


MLMODEL

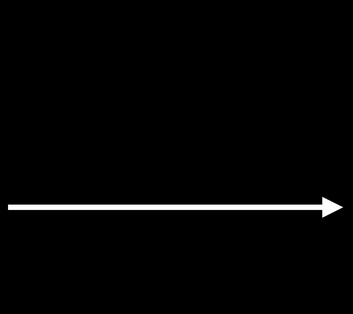
Umbrella

Beach

Chair

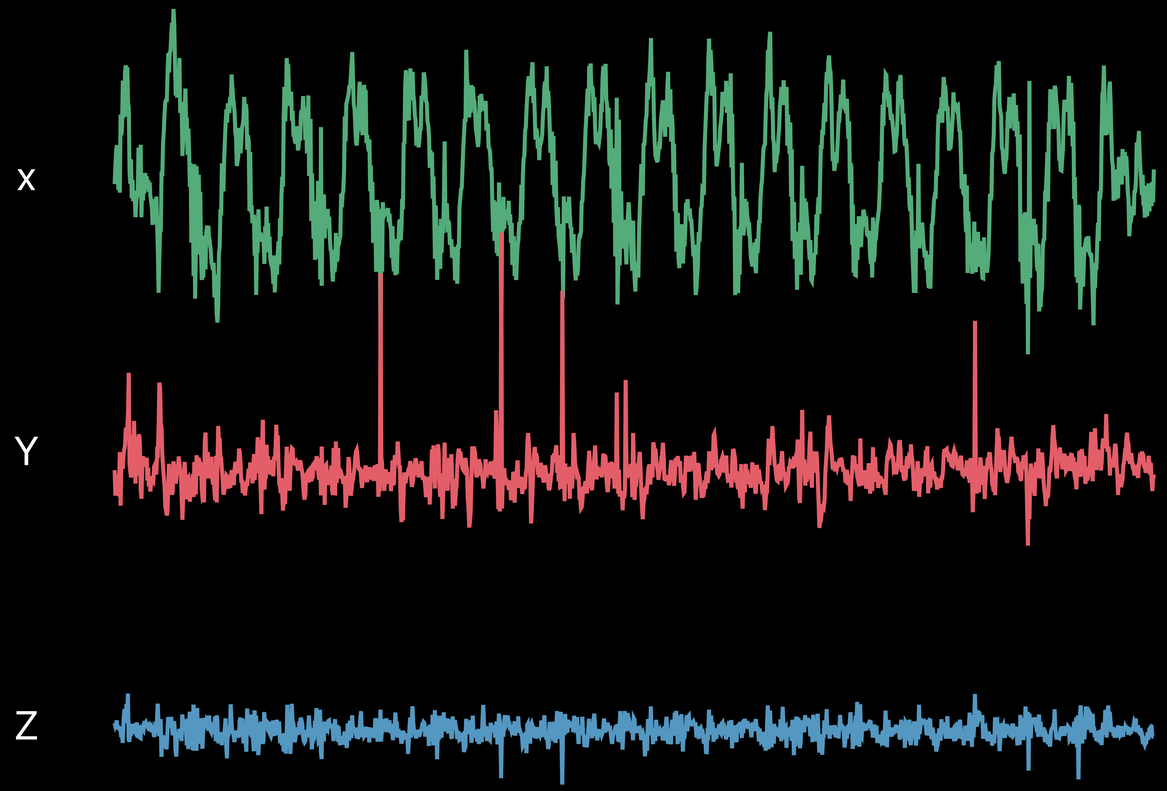
Chair

“Keep up the great work Cate!”



Motivation

Jazz

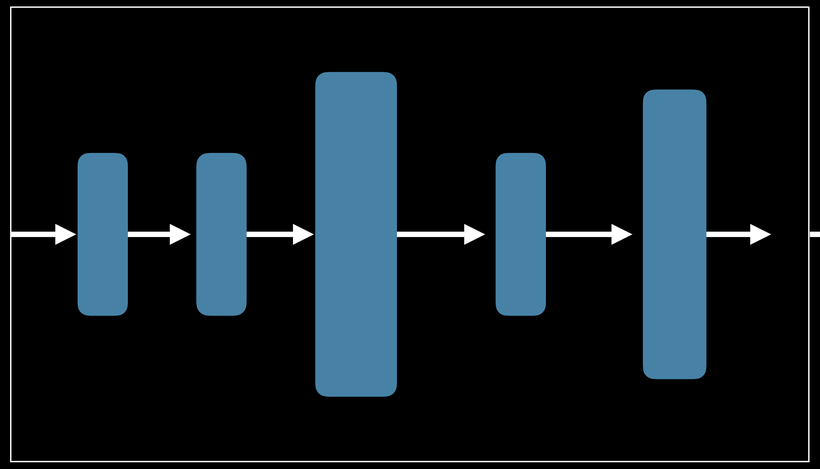


Rowing Action

Functionality Encoded in a Model

Beach

Learned From Data



Beach

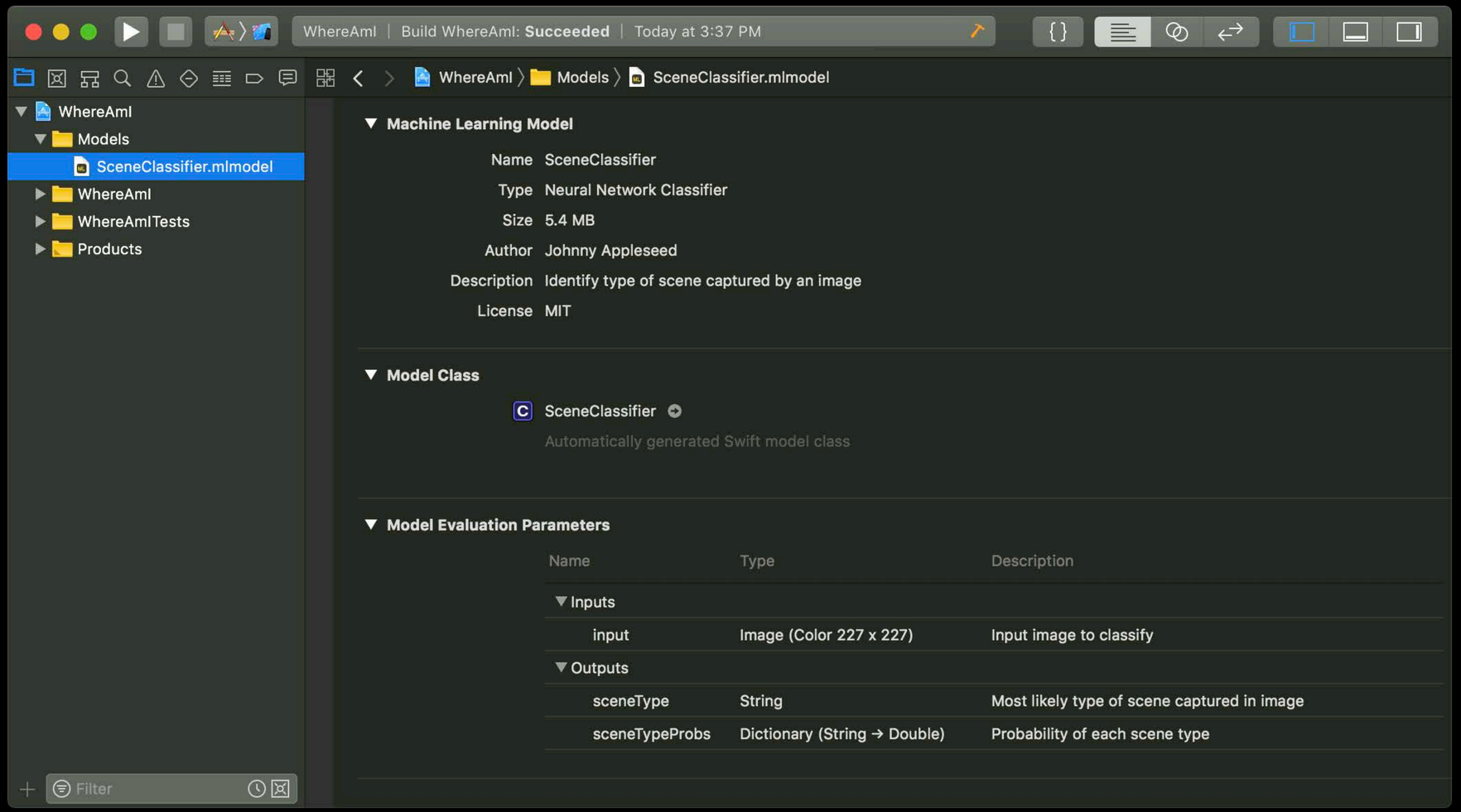
Neural Network Tree Ensemble

Linear Model Support Vector Machine

Simplified

Beach

Xcode


WhereAml | Build WhereAml: Succeeded | Today at 3:37 PM
WhereAml 〉 Models 〉 SceneClassifier.mlmodel
WhereAml
Models
SceneClassifier.mlmodel
WhereAml
WhereAmlTests
Products
Filter
Machine Learning Model
Name SceneClassifier
Type Neural Network Classifier
Size 5.4 MB
Author Johnny Appleseed
Description Identify type of scene captured by an image
License MIT
Model Class
SceneClassifier
Automatically generated Swift model class
Model Evaluation Parameters
Name
Type
Description
Inputs
input
Image (Color 227 x 227)
Input image to classify
Outputs
sceneType
String
Most likely type of scene captured in image
sceneTypeProbs
Dictionary (String → Double)
Probability of each scene type

```
let model = SceneClassifier()

let prediction = try model.prediction(input: image)

return prediction.sceneType
```

```
let model = SceneClassifier()

let prediction = try model.prediction(input: image)

return prediction.sceneType
```

```
let model = SceneClassifier()
let prediction = try model.prediction(input: image)
return prediction.sceneType
```

```
let model = SceneClassifier()

let prediction = try model.prediction(input: image)

return prediction.sceneType
```

Integrated

Vision

`VNCoreMLRequest`

Natural Language

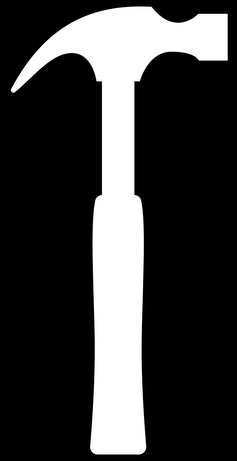
`NLModel`

What’s New?

What's New?

Part One

Part Two



Part One

Part One

Model Size

Part One

Model Size

Performance

Part One

Model Size

Performance

Customization

Model Size

Francesco Rossi, Core ML

Model Size

Performance

Customization

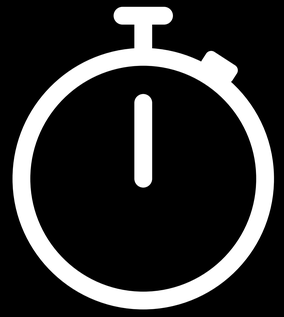
Models on Device

Models on Device

Privacy

Models on Device

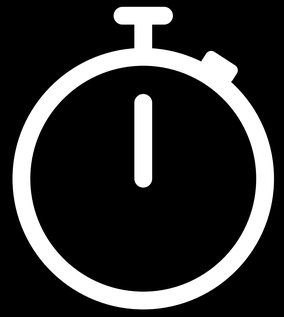
Privacy



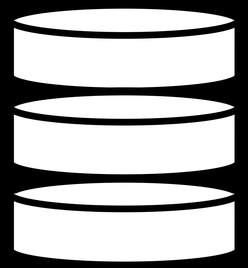
Speed

Models on Device

Privacy



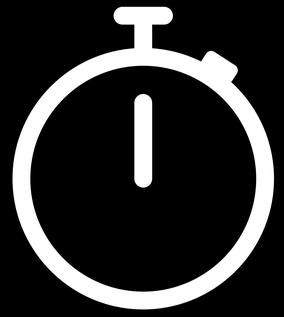
Speed



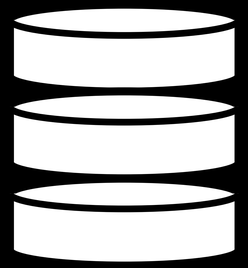
No Server

Models on Device

Privacy



Speed

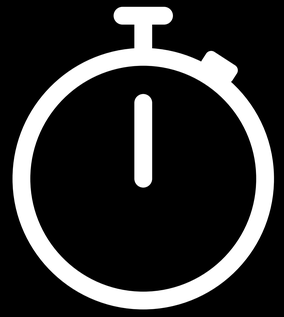


No Server

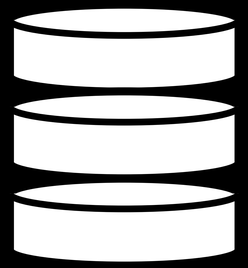
Available

Models on Device

Privacy



Speed



No Server

Available

Need to store machine learning models on device

Awesomeness

Awesomeness

More
Awesomeness

More
Awesomeness

App Size: **+100 MB**

More
Awesomeness

App Size: **+100 MB**

Even More
Awesomeness

App Size: **+250 MB**

Download on Demand

Compile on device

More
Awesomeness

Initial Install Size: **Unchanged**

Download on Demand

Compile on device

More
Awesomeness

Initial Install Size: **Unchanged**
On demand: +**250** MB

Download on Demand

Compile on device

More
Awesomeness

Initial Install Size: **Unchanged**
On demand: **+250** MB

Reducing Model Size

Reducing Model Size

Smaller bundle

Reducing Model Size

Smaller bundle

Smaller/faster downloads

Reducing Model Size

Smaller bundle

Smaller/faster downloads

Reduced runtime memory usage

Core ML App Size

Factors

Core ML App Size

Factors

Number of Models

Core ML App Size

Factors

Number of Models x **Number of Weights**

Core ML App Size

Factors

Number of Models x **Number of Weights** x **Size of Weight**

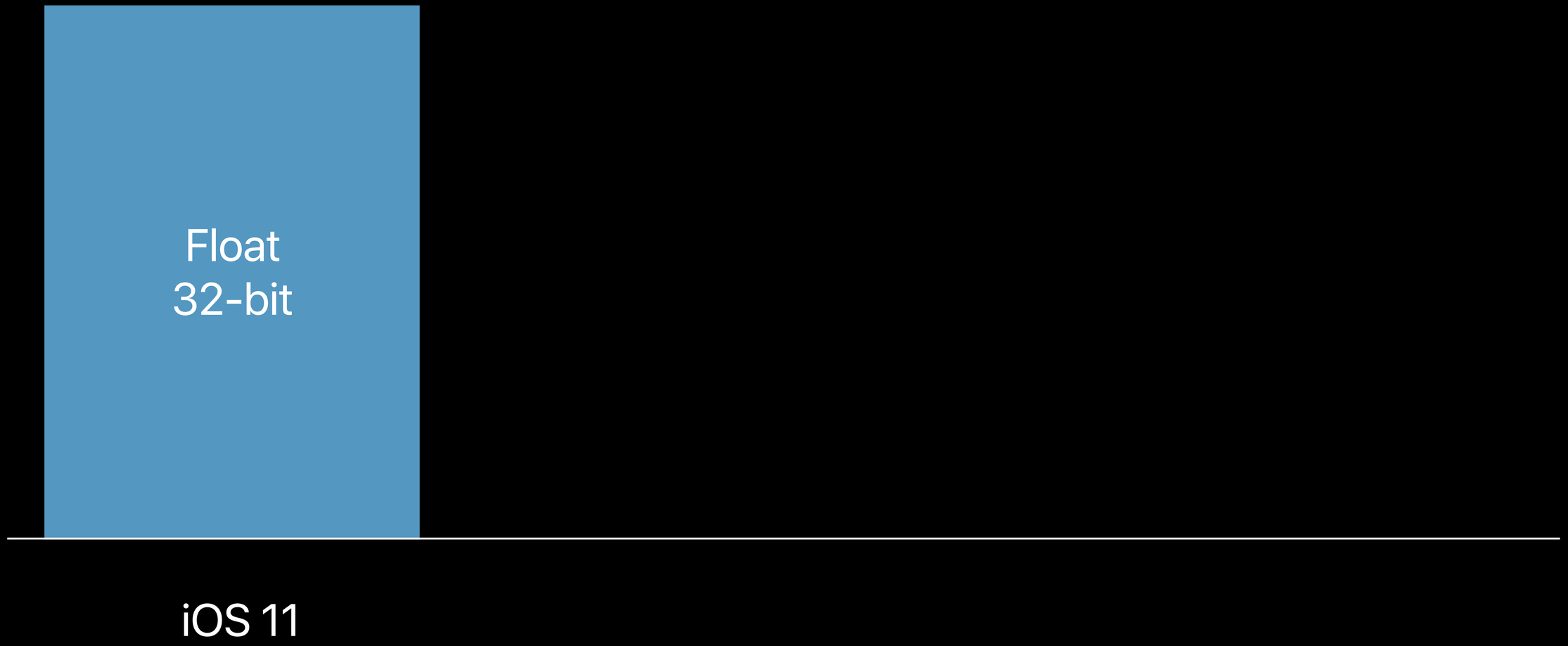
Core ML App Size

Factors

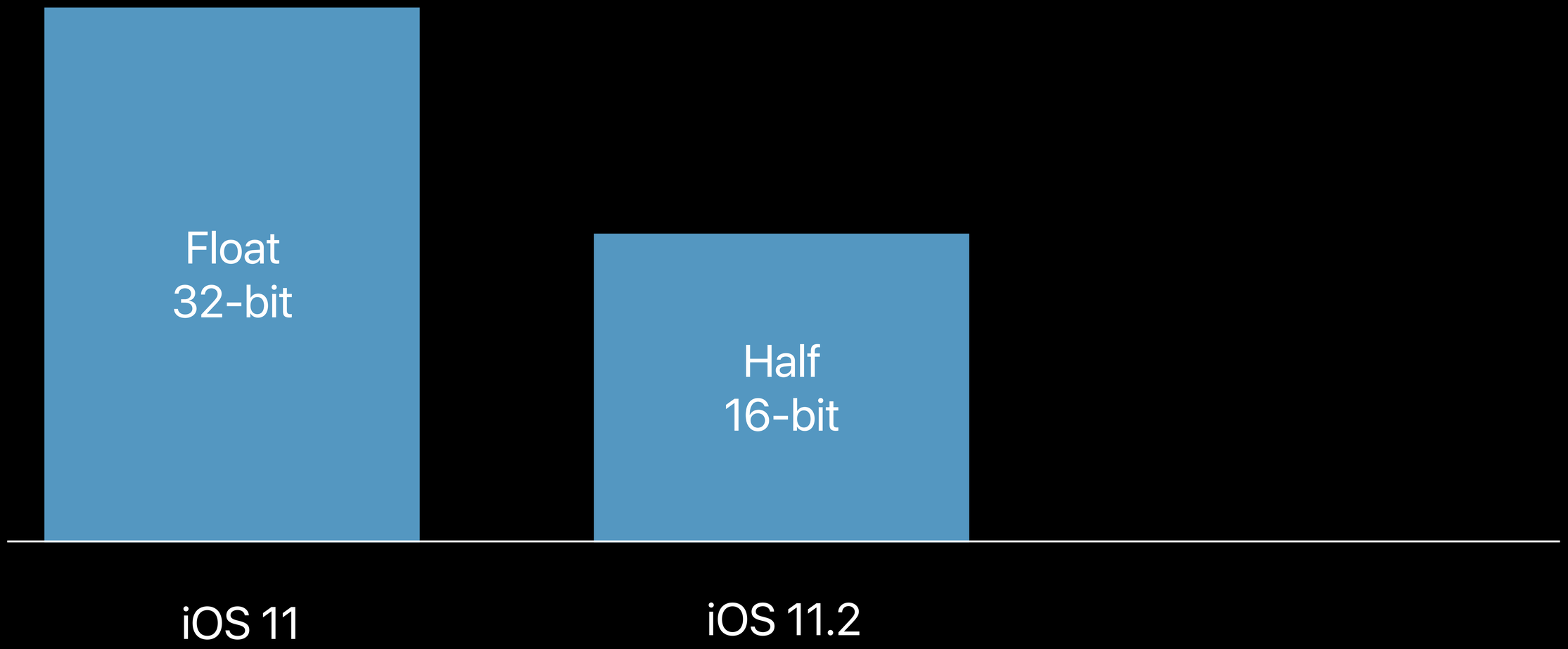
Number of Models x **Number of Weights** x **Size of Weight**

Options for Neural Networks

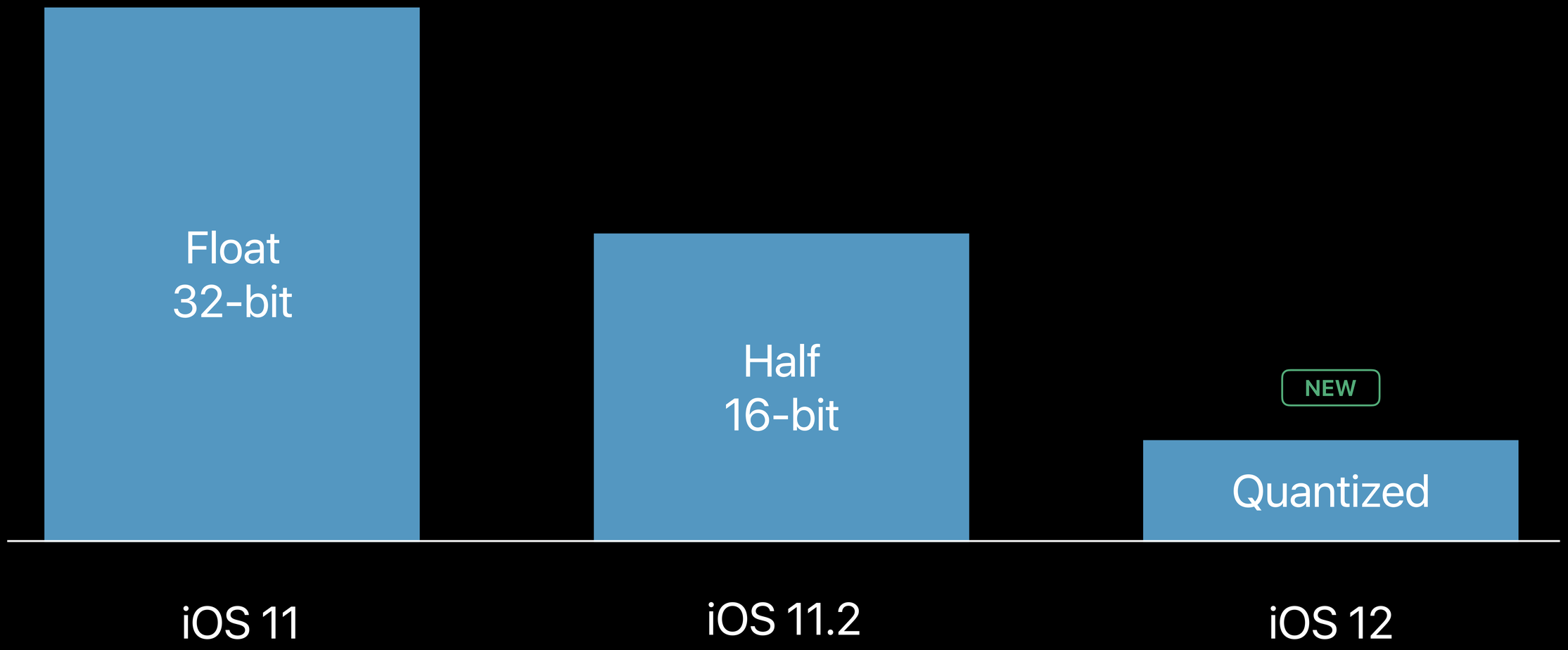
Options for Neural Networks



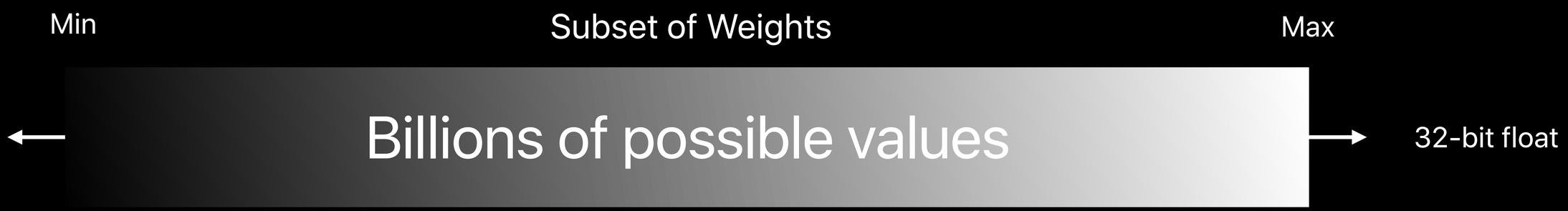
Options for Neural Networks



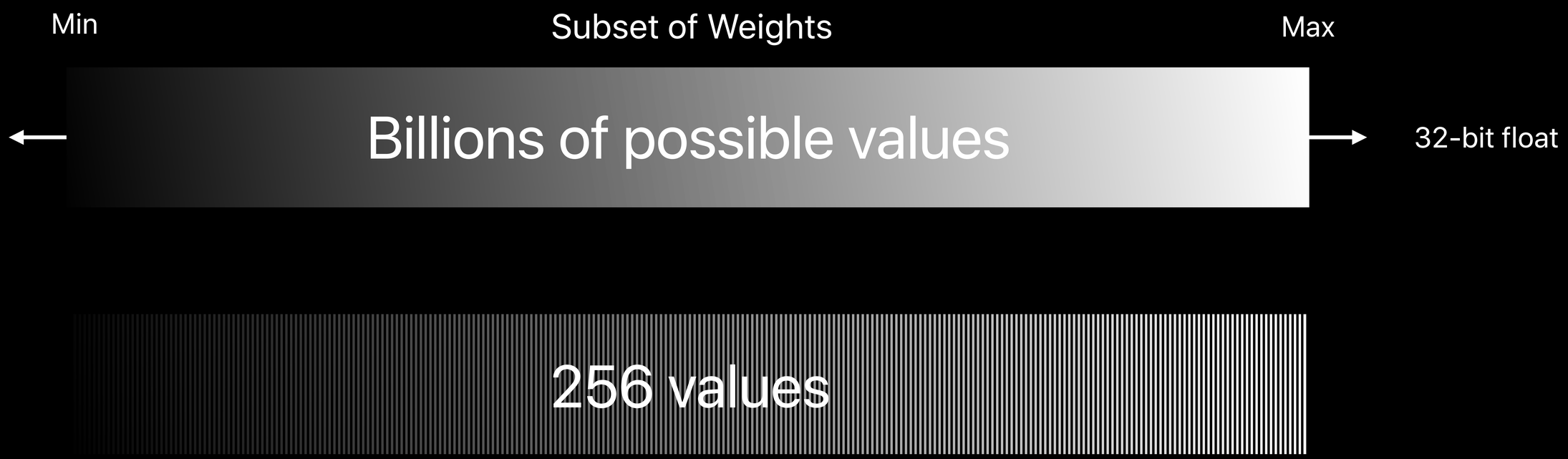
Options for Neural Networks



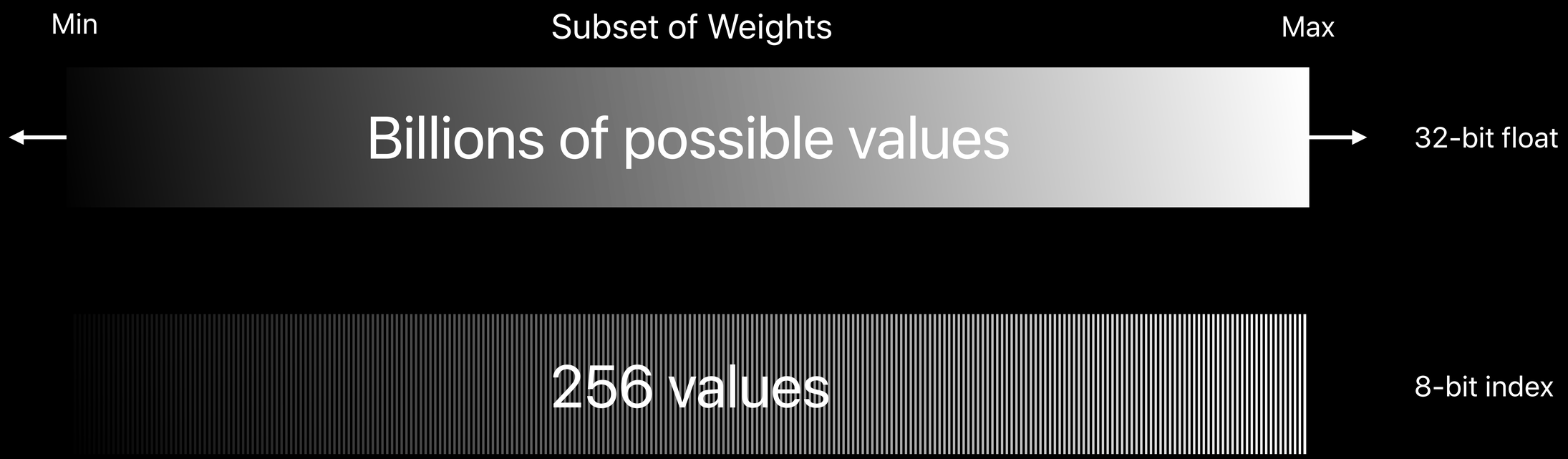
Weight Quantization



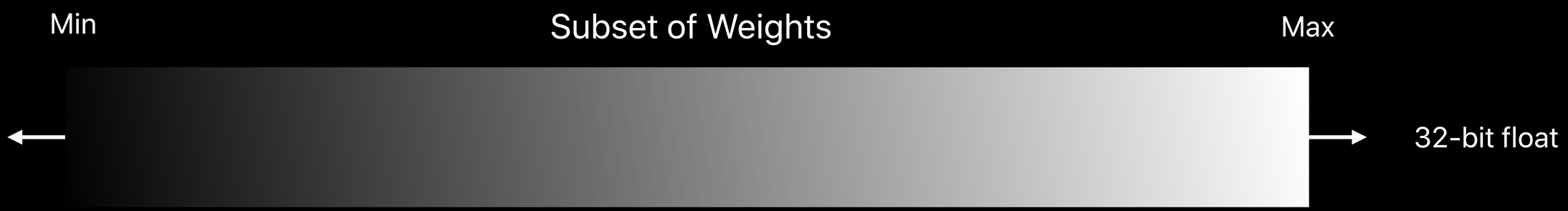
Weight Quantization



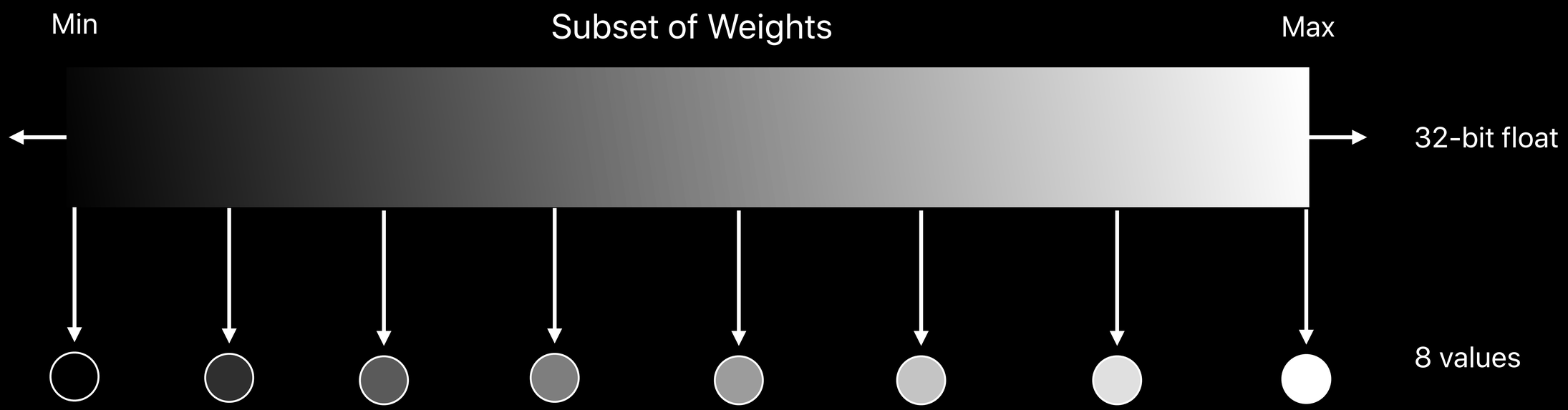
Weight Quantization



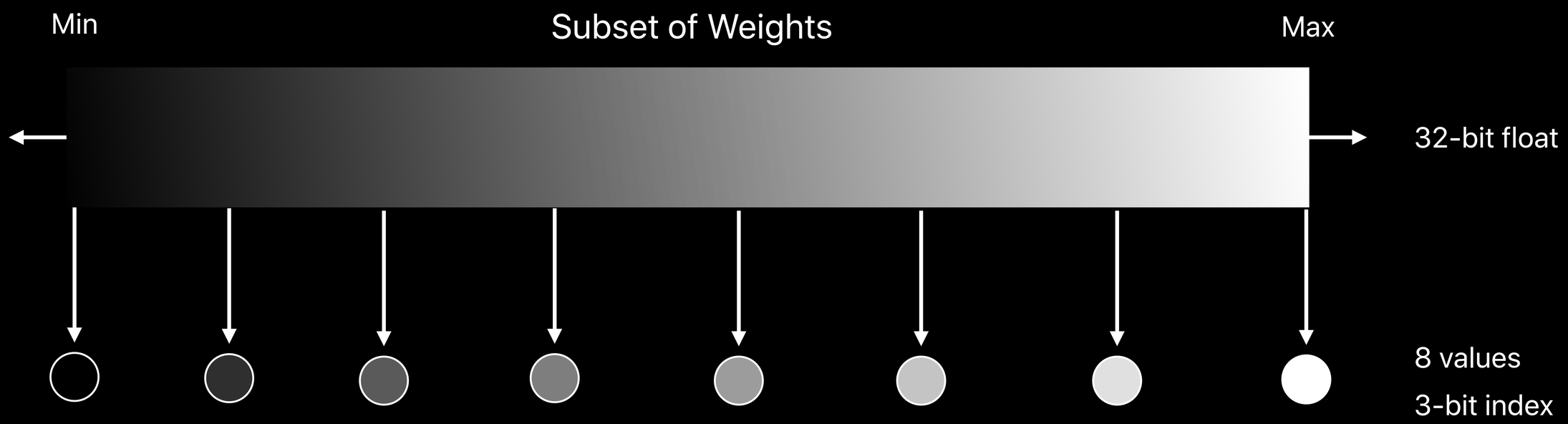
Weight Quantization



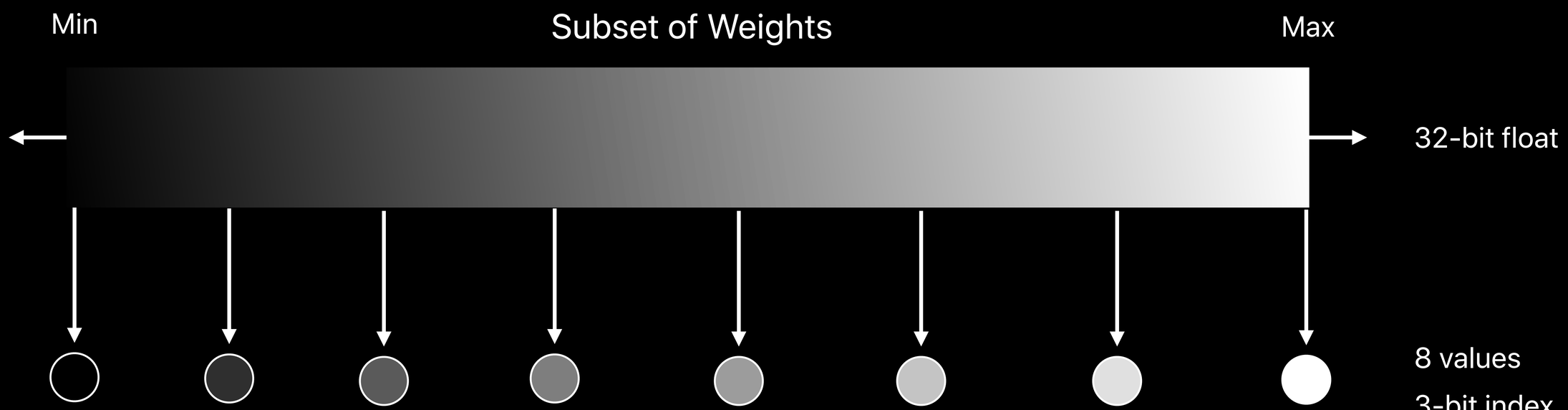
Weight Quantization



Weight Quantization



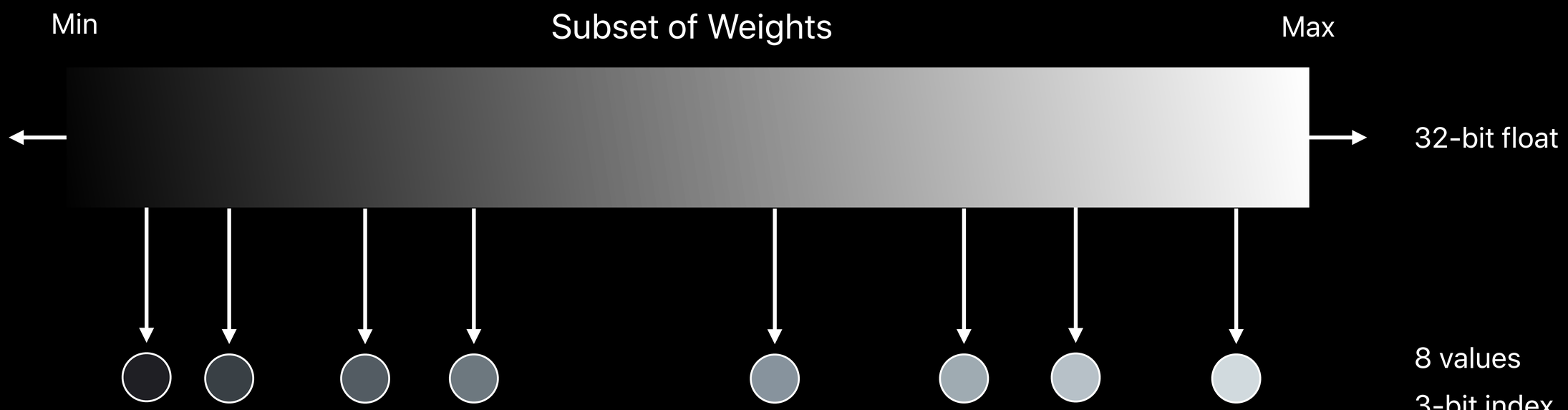
Weight Quantization



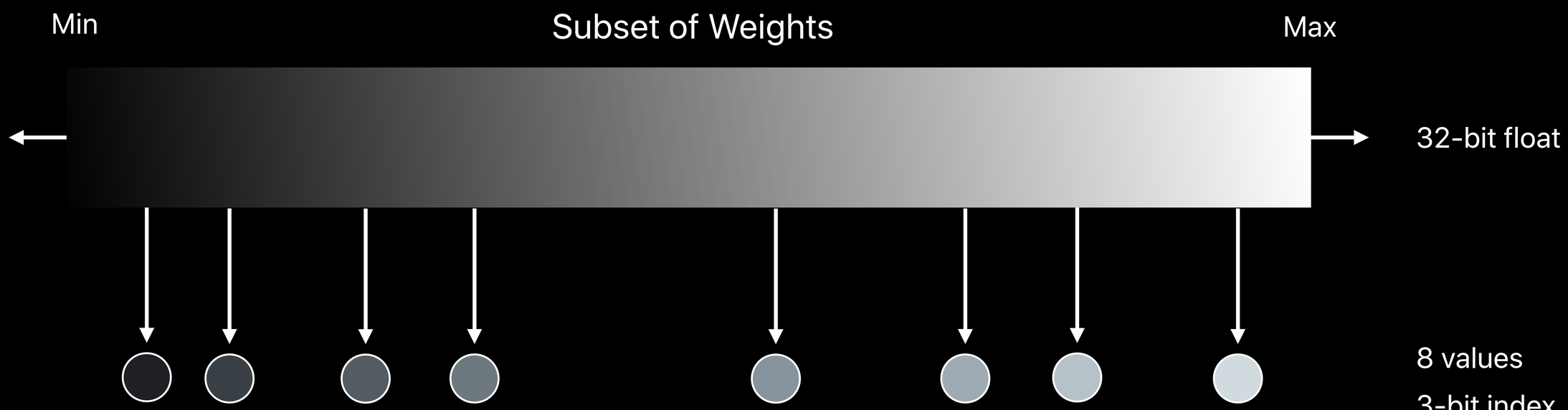
Linear

Uniform distribution

Weight Quantization



Weight Quantization



Lookup Table

Non-uniform distribution

Resnet50

25 Million Weights
32-bit Float

103 MB

Resnet50

8 bit Quantization

25 Million Weights
32-bit Float

103 MB

25 Million Weights
8-bit UInt8

26 MB

Resnet50

< 8 bit quantization

25 Million Weights
32-bit Float

103 MB

25 Million Weights
8-bit UInt8

26 MB

25 Million Weights
4-bit UInt4

14 MB

Resnet50

< 8 bit quantization

25 Million Weights
32-bit Float

103 MB

25 Million Weights
8-bit UInt8

26 MB

25 Million Weights
4-bit UInt4

14 MB

All the way down to 1-bit

Obtaining Quantized Models

Obtaining Quantized Models

Post-training quantization

Obtaining Quantized Models

Post-training quantization

Train quantized

- From scratch or re-training
- Then convert quantized models to Core ML

Accuracy Tradeoff

Model dependent

Use case dependent

Active area of research

Accuracy Tradeoff

Model dependent

Use case dependent

Active area of research

Check Your Quantized Model Accuracy

Accuracy Tradeoff

Model dependent

Use case dependent

Active area of research

Check Your Quantized Model Accuracy

With test data and metric relevant for your app

Demo

Demo Summary

4 Models

Demo Summary

4 Models

6.7 MB each
32-bit

26.8 MB

Demo Summary

4 Models

6.7 MB each
32-bit

857 KB each
4-bit

26.8 MB

3.4 MB

32-bit

6.7 MB

8-bit

1.7 MB

4-bit

857 KB

Factors

Number of Models x **Number of Weights** x **Size of Weight**

Factors

Number of Models x **Number of Weights** x **Size of Weight**

Number of Models

Function A

Function B

Function C

Number of Models

Function A

Function B and C

Number of Models

Function A

Function B and C

Multi-task models

Example: See Turi Create session

Number of Models

Function A

Function B and C

Multi-task models
Example: See Turi Create session

Flexible shapes
and sizes

Number of Models

Function A

Function B and C

Multi-task models
Example: See Turi Create session

Flexible shapes and sizes

Style Transfer Model

Input size defined by the model

GlassSD.mlmodel

Name	Type	Description
▼ Inputs		
image	Image (Color 640 x 480)	Input image to stylize
▼ Outputs		
stylized	Image (Color 640 x 480)	Stylized image

Same style for different sizes??

Input Size Mismatch

1920 x 1440

Glass.mlmodel

640 x 480

Input Size Mismatch

1920 x 1440

Glass.mlmodel

640 x 480

Input Size Mismatch

1920 x 1440

Glass.mlmodel

640 x 480

Input Size Mismatch

Model for High Resolution Input

1920 x 1440

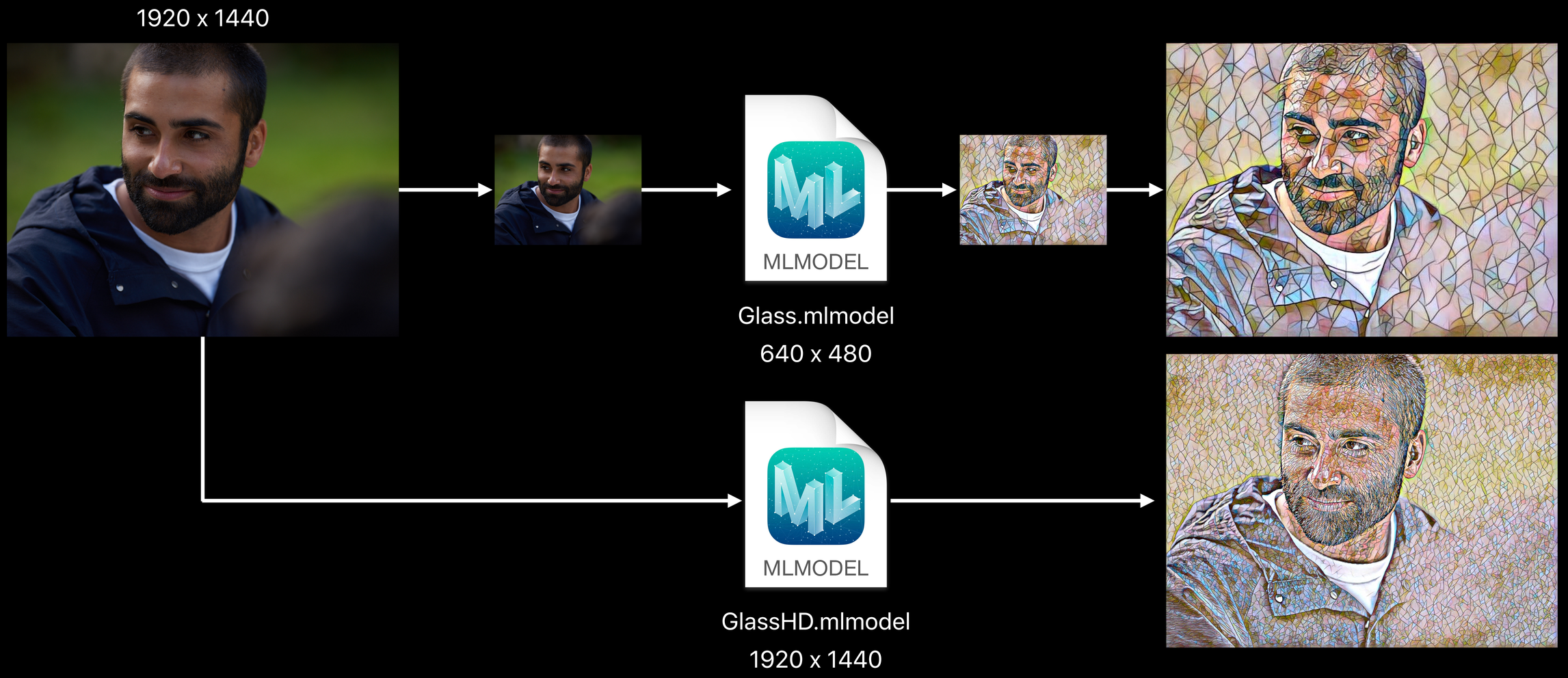
Glass.mlmodel

640 x 480

GlassHD.mlmodel

1920 x 1440

Model for High Resolution Input



Model Per Resolution

640 x 480

Glass.mlmodel

1920 x 1440

GlassHD.mlmodel

Model Per Resolution

640 x 480

Glass.mlmodel

1920 x 1440

GlassHD.mlmodel

2x Size

One Flexible Model

640 x 480 or 1920 x 1440

GlassFlexible.mlmodel

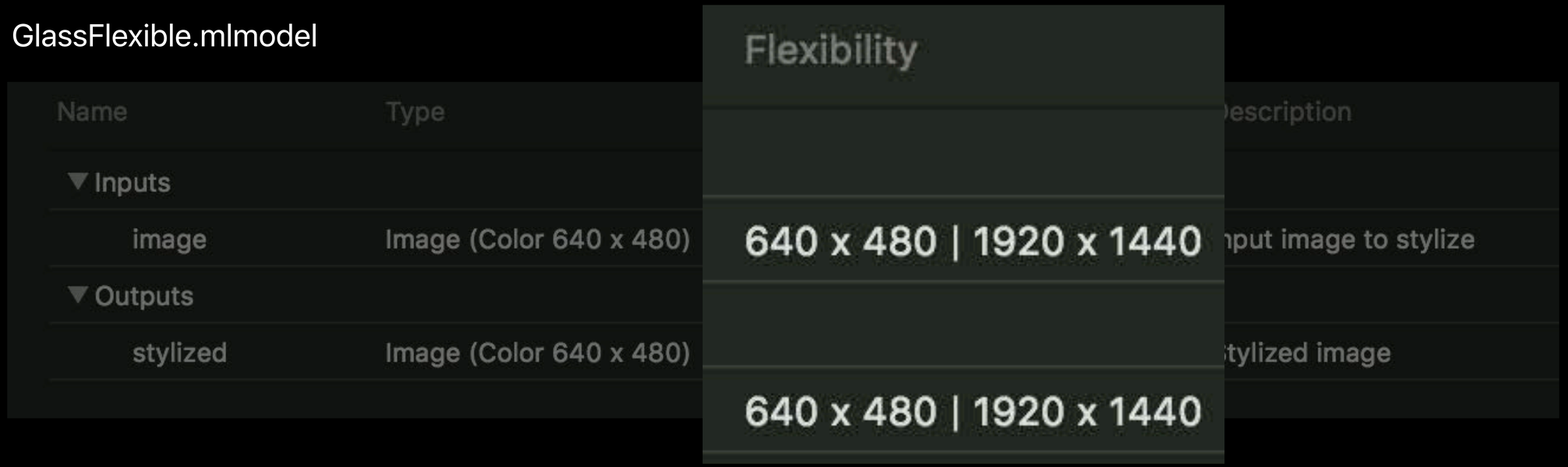
Combine Using Flexible Image Sizes

GlassFlexible.mlmodel

Name	Type	Flexibility	Description
▼ Inputs			
image	Image (Color 640 x 480)	640 x 480 | 1920 x 1440	Input image to stylize
▼ Outputs			
stylized	Image (Color 640 x 480)	640 x 480 | 1920 x 1440	Stylized image

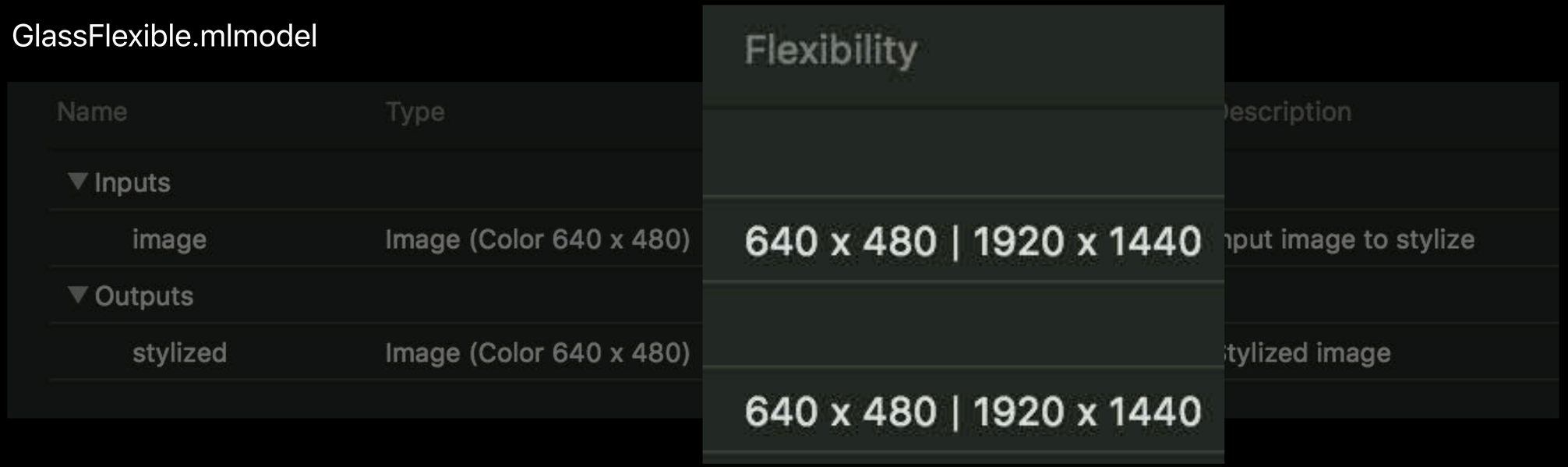
Combine Using Flexible Image Sizes

GlassFlexible.mlmodel



Combine Using Flexible Image Sizes

GlassFlexible.mlmodel



One model

No redundant code

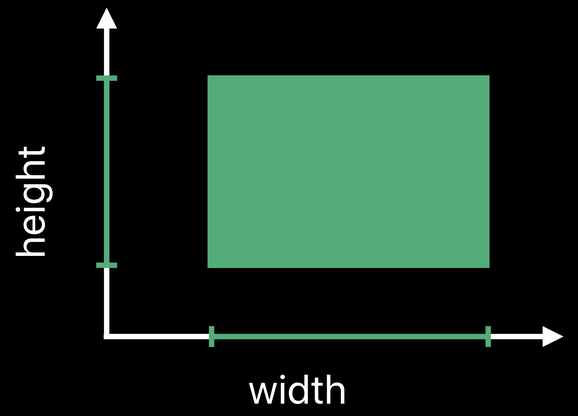
Faster model switching times

Flexibility Options

Flexibility Options

Size Range Per Dimension

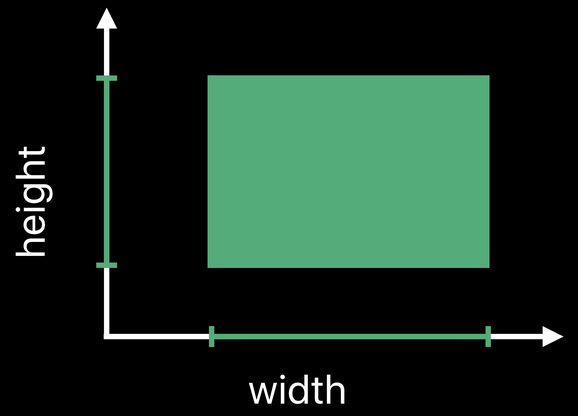
- Flexibility



Flexibility Options

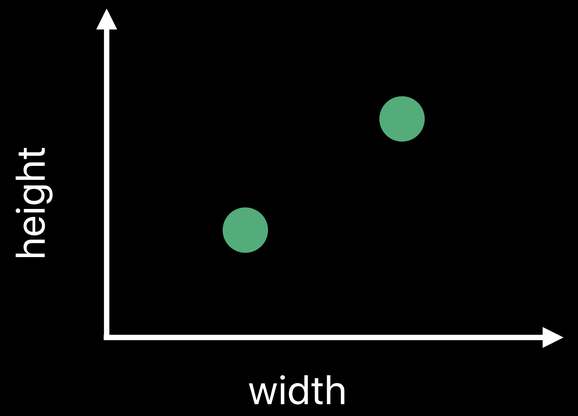
Size Range Per Dimension

- Flexibility



Enumerated

- Better performance
- Smaller testing surface



Which Models are Flexible?

Which Models are Flexible?

Fully Convolutional Neural Networks

- Image processing
- Object detection

Core ML Tools can check for you!

Model Size

Number of Models x **Number of Weights** x **Size of Weight**

Flexible Sizes

Quantization

Model Size

Number of Models	x	Number of Weights	x	Size of Weight
Flexible Sizes		Broad ML Support		Quantization

Performance and Customization

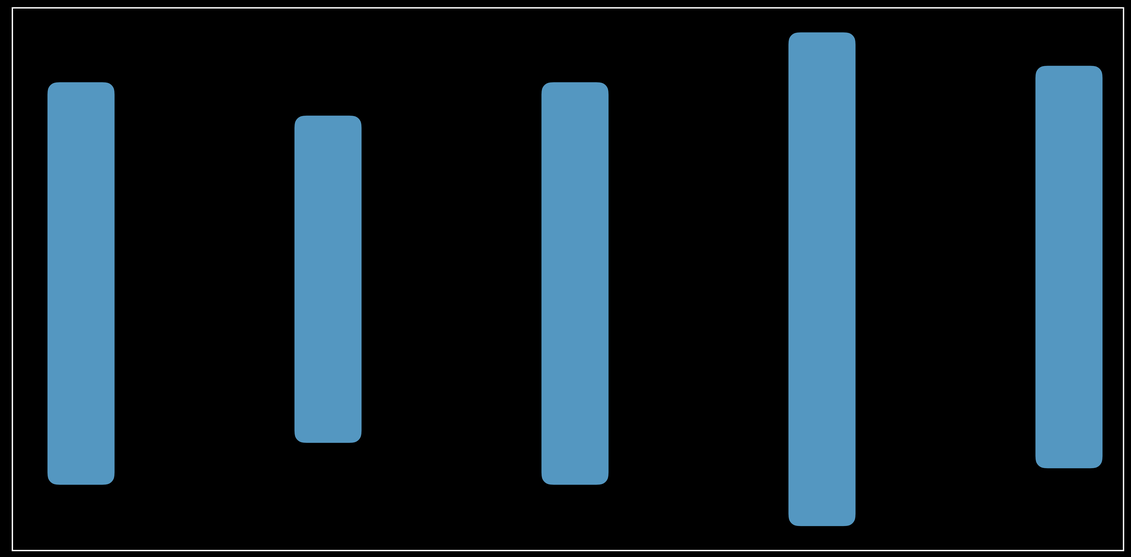
Bill March, Core ML

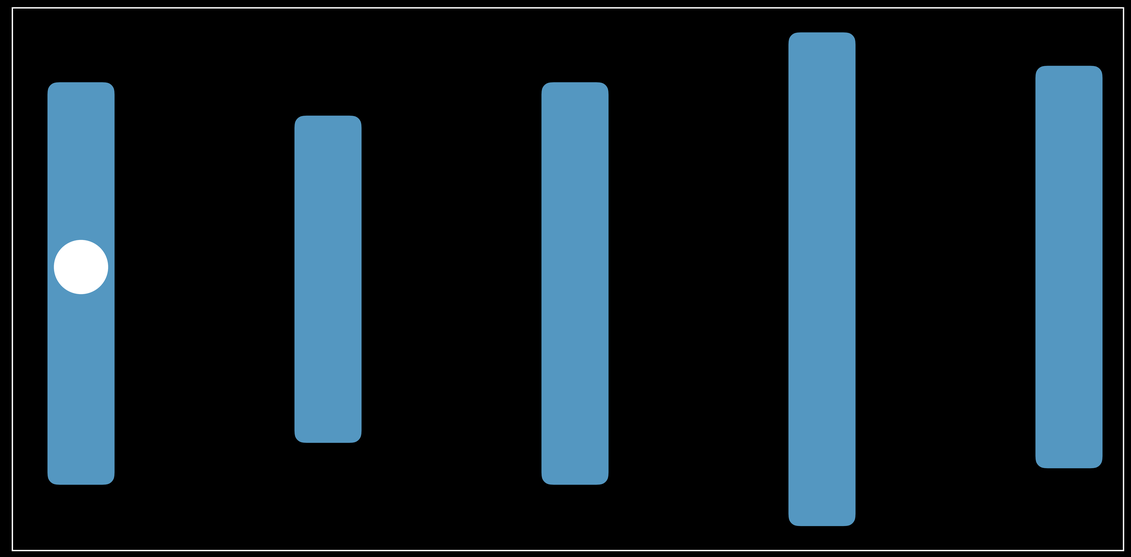
Model Size

Performance

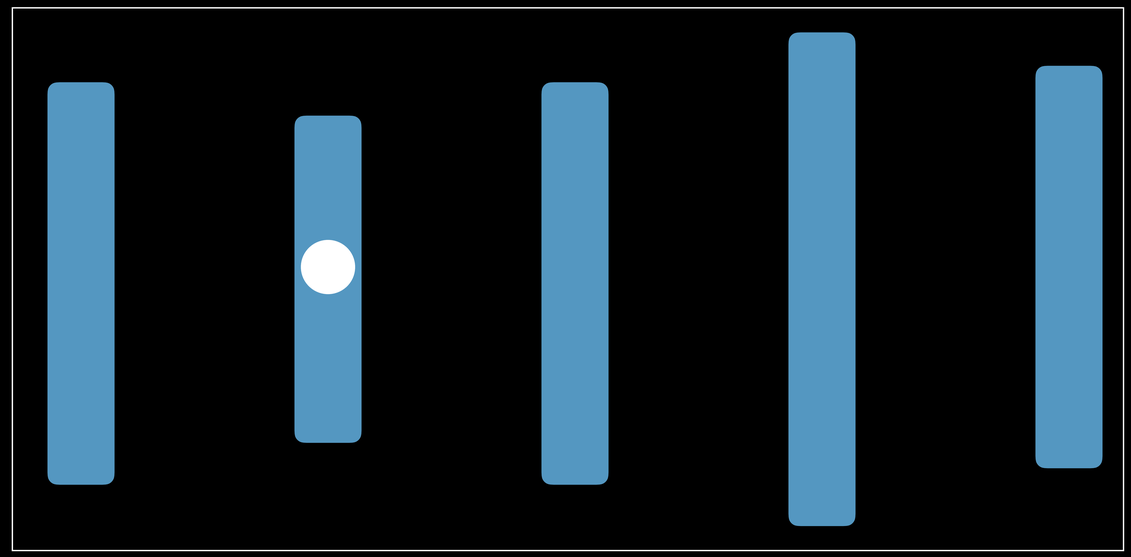
Customization


MLMODEL

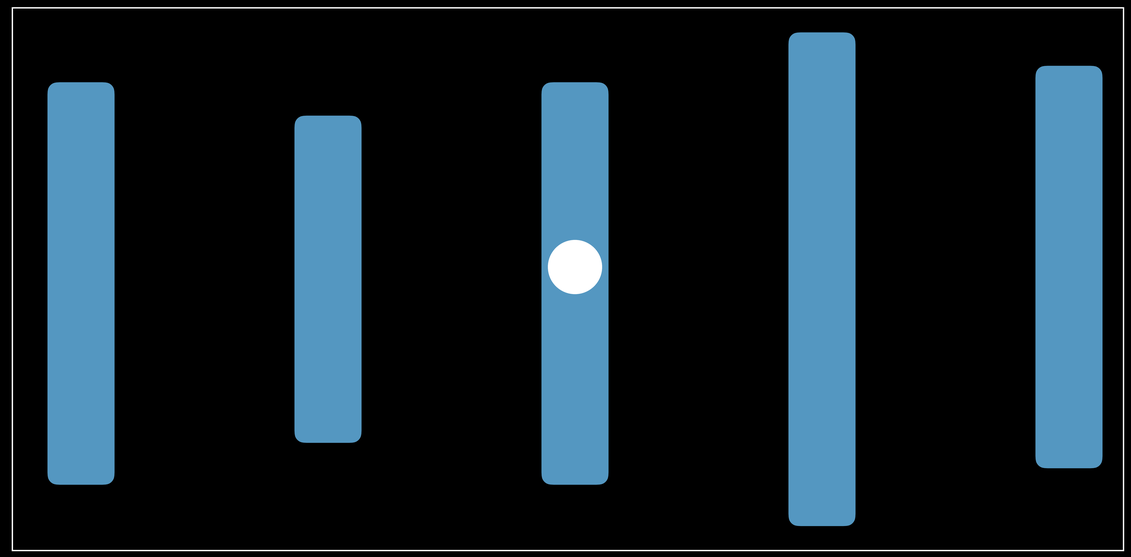




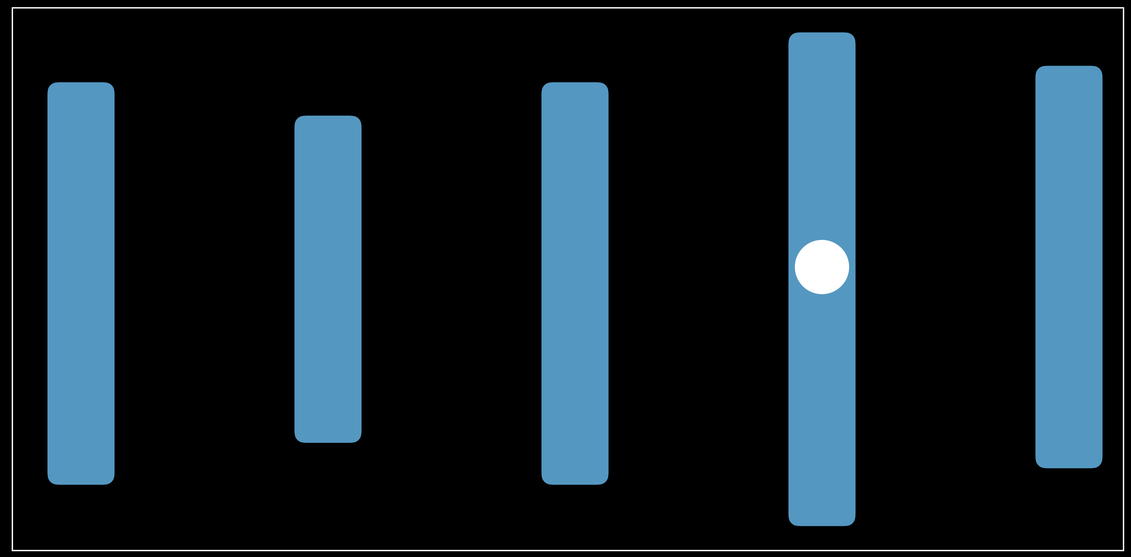
GPU



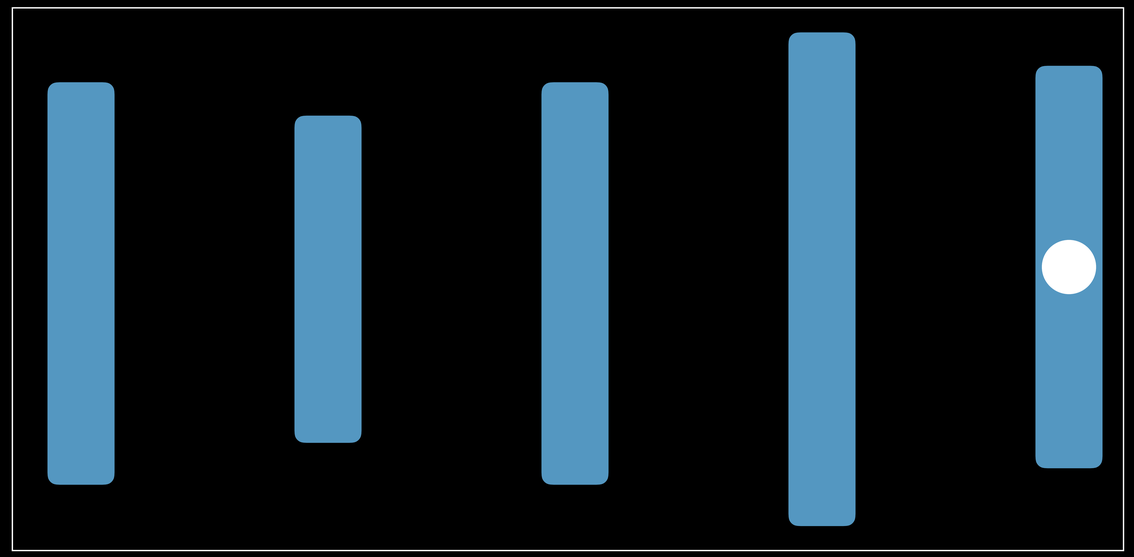
GPU



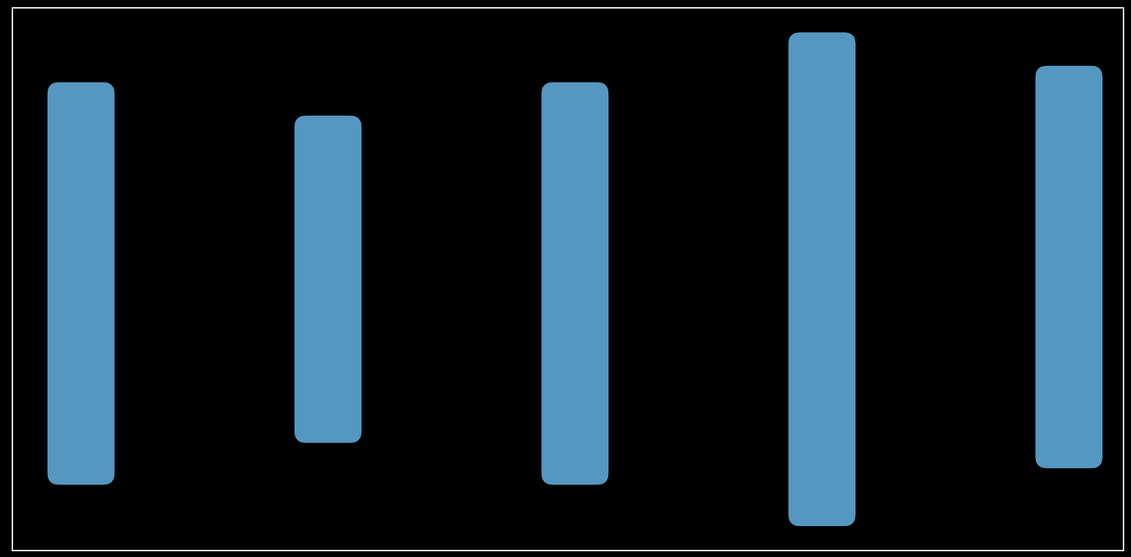
CPU



GPU

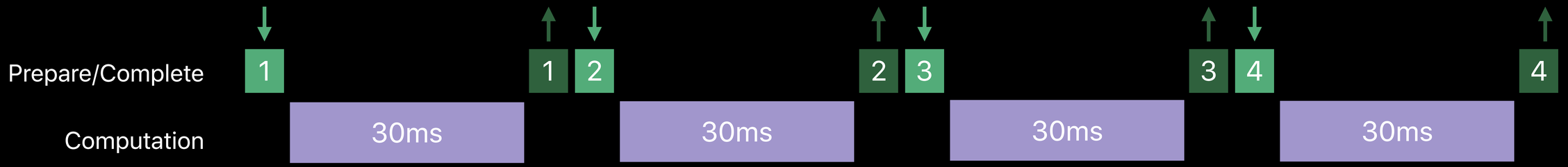


CPU

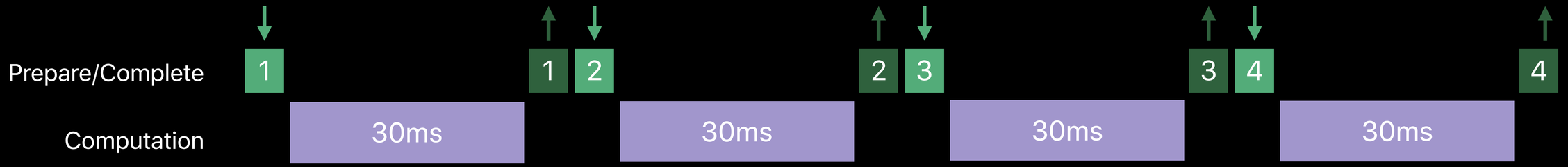


```
// Loop over inputs
for i in 0..< modelInputs.count {
    modelOutputs[i] = model.prediction(from: modelInputs[i], options: options)
}
```

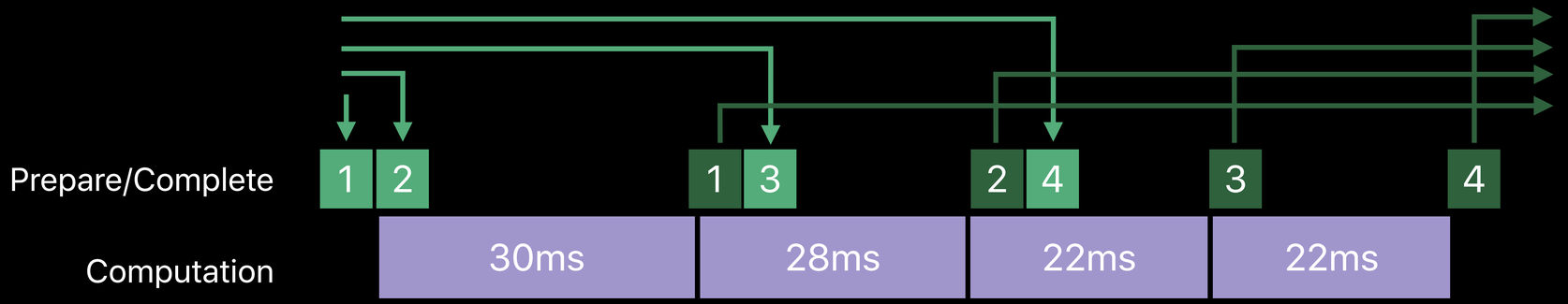
For Loop


Prepare/Complete
Computation
1
1
2
2
3
3
4
4
30ms
30ms
30ms
30ms

For Loop



Batch

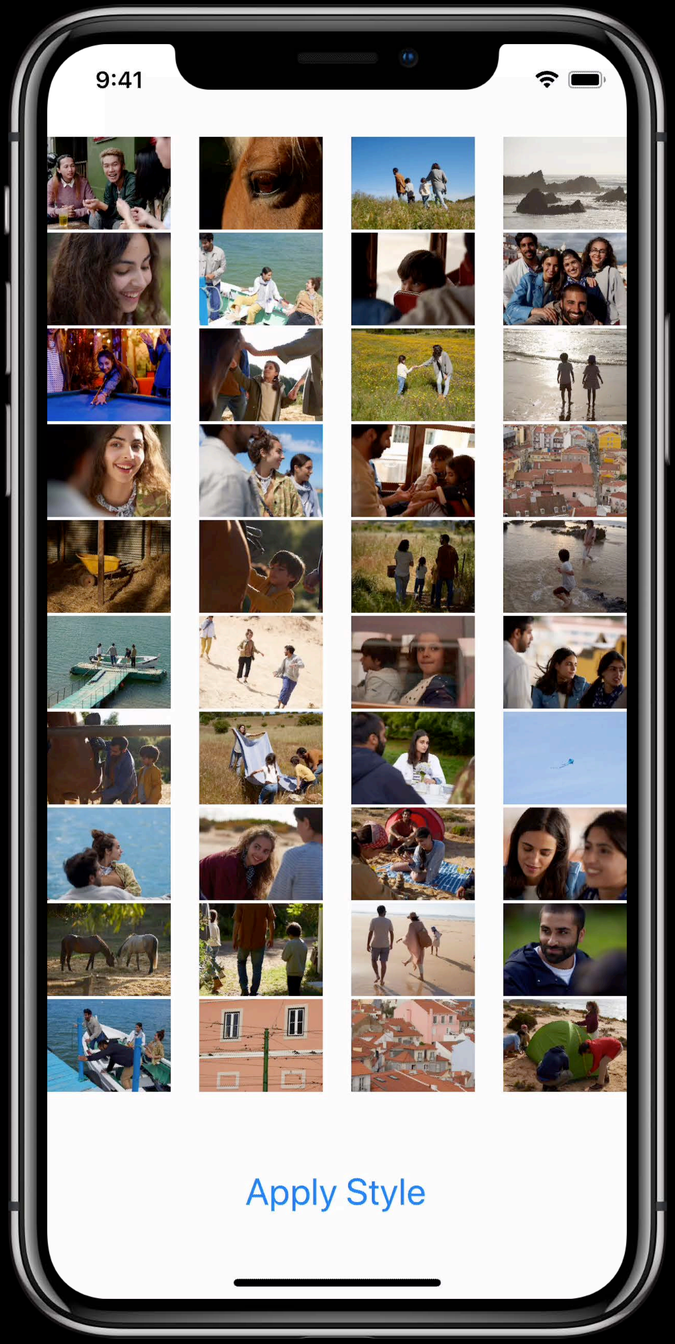


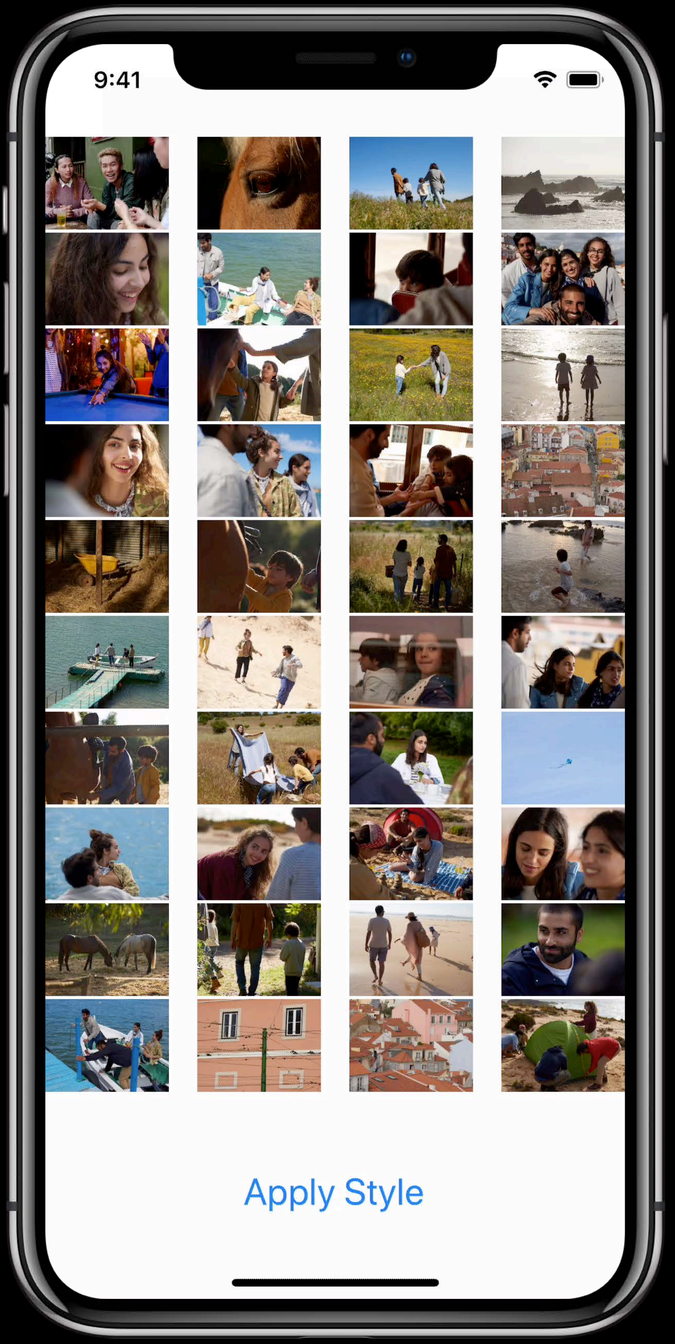

For Loop
Batch

```
// Loop over inputs
for i in 0..< modelInputs.count {
    modelOutputs[i] = model.prediction(from: modelInputs[i], options: options)
}
```

```
modelOutputs = model.predictions(from: modelInputs, options: options)
```

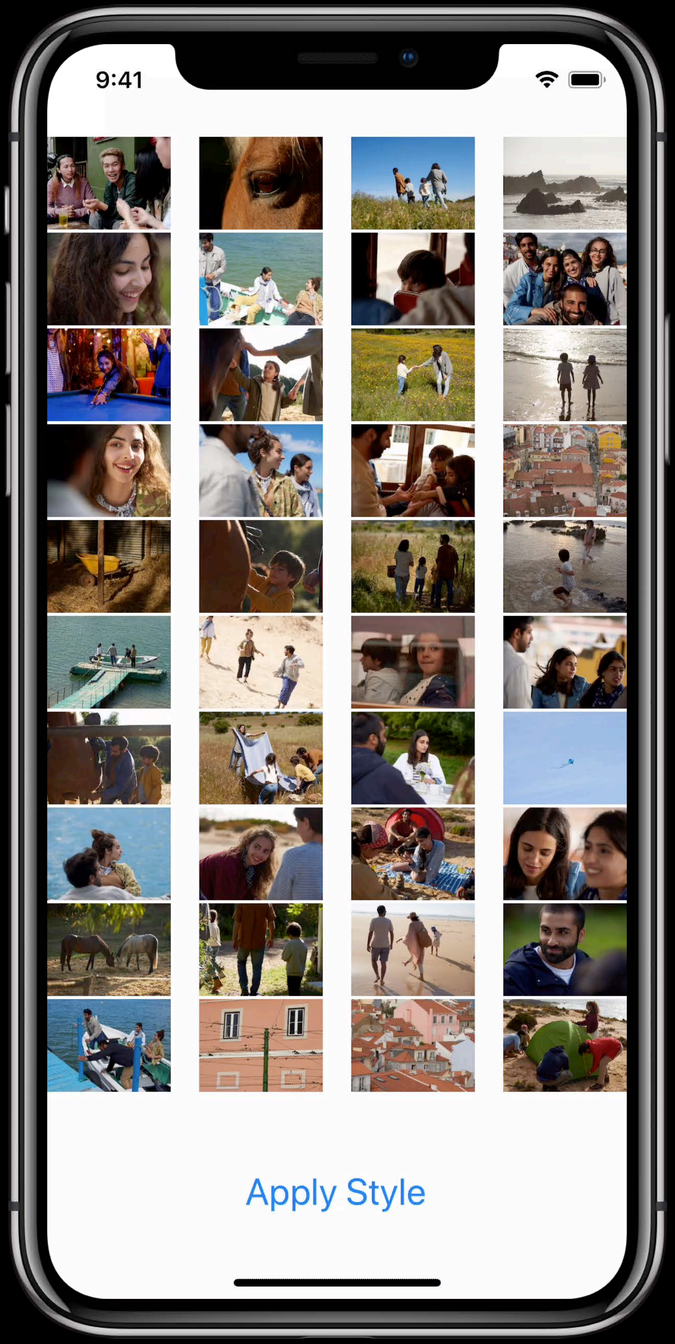
Loop

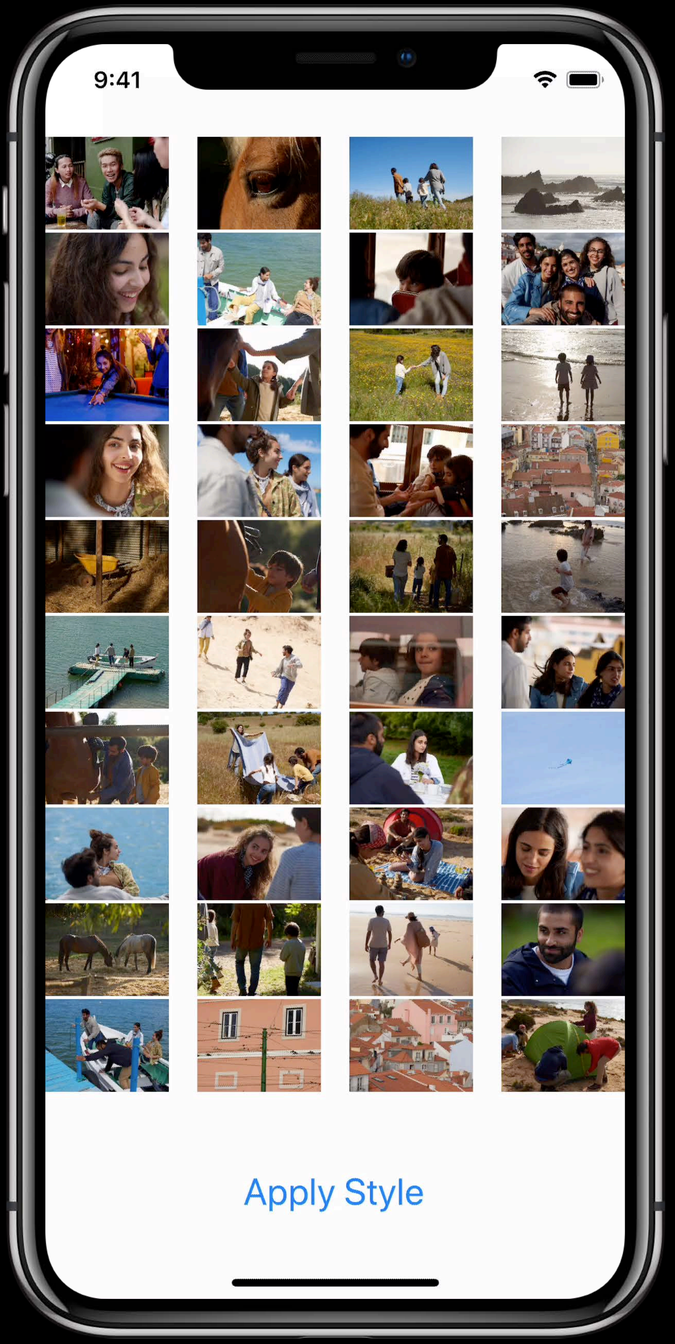




Batch

Loop





Batch

Model Size

Performance

Customization

Any Horses?

Any Horses?

~2007

Hope computers can answer some day, but for now—just ask someone.

Any Horses?

~2012

Wow—Computers *can* answer this!
Cutting-edge deep learning research is exciting.

Any Horses?

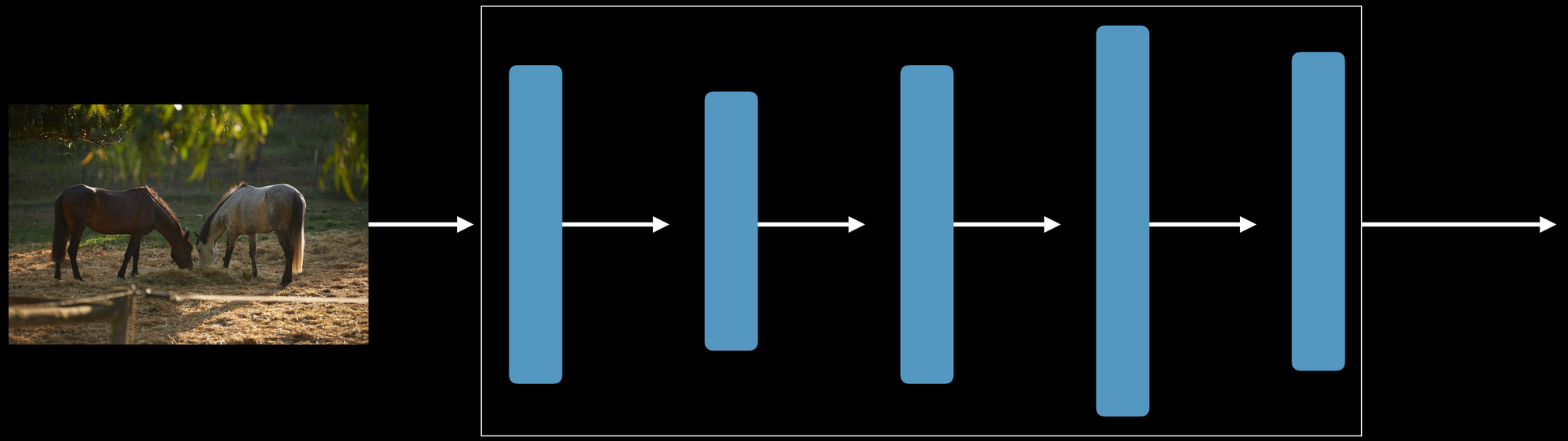
2018

Train a model with Create ML—the UI is great!

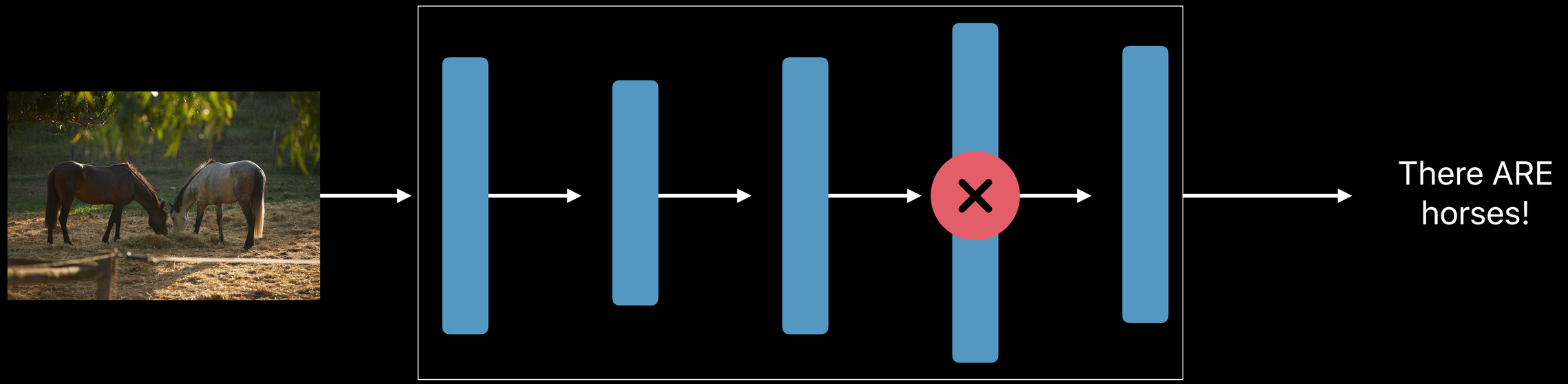
MLMODEL

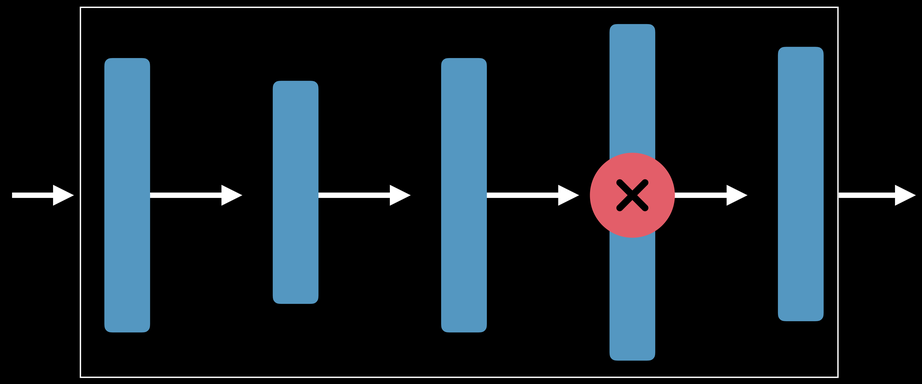
There ARE horses!

Custom Layers



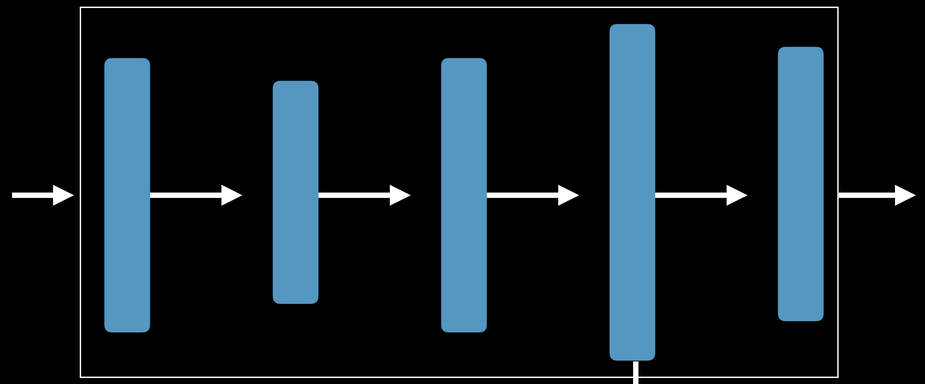
Custom Layers





Custom Layers

iOS 11.2




```
@objc(AAPLCustomHorseLayer)
class AAPLCustomHorseLayer: MLCustomLayer {
    . . .
}
```

```
public protocol MLCustomLayer {

    public init(parameters: [String : Any]) throws

    public func setWeightData(_ weights: [Data]) throws

    public func outputShapes(forInputShapes: [[NSNumber]]) throws -> [[NSNumber]]

    public func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws
}
```

```
public protocol MLCustomLayer {

    public init(parameters: [String : Any]) throws

    public func setWeightData(_ weights: [Data]) throws

    public func outputShapes(forInputShapes: [[NSNumber]]) throws -> [[NSNumber]]

    public func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws
}
```

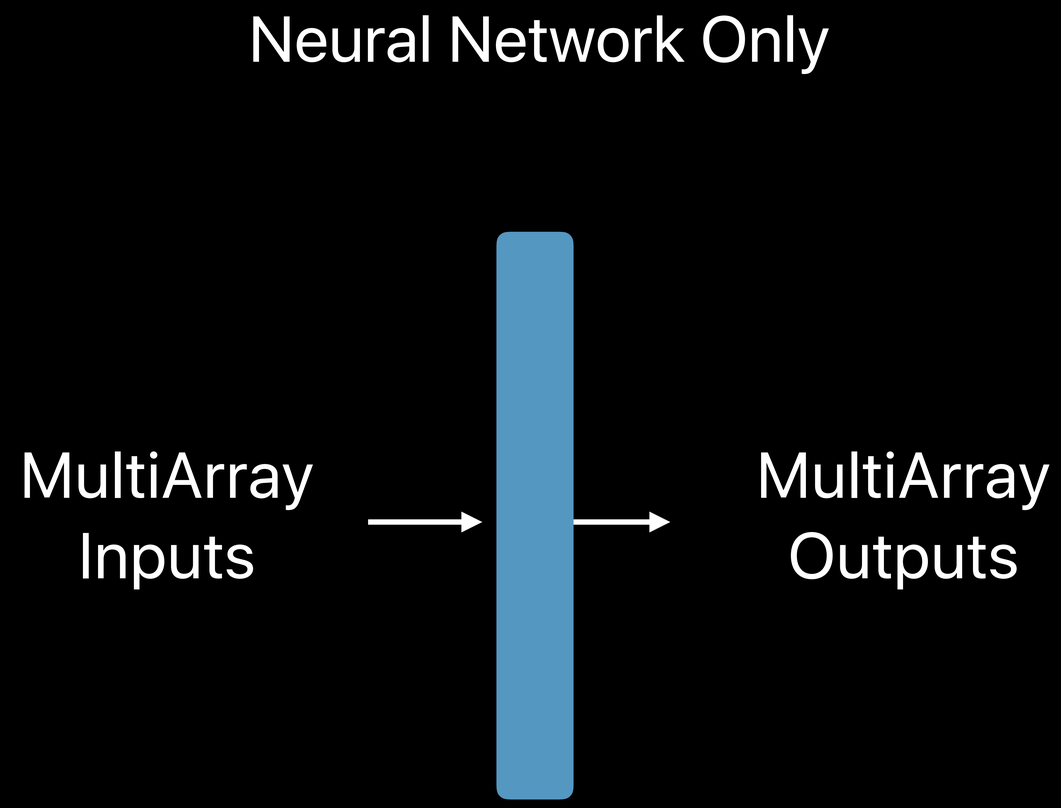
```
public protocol MLCustomLayer {

    public init(parameters: [String : Any]) throws

    public func setWeightData(_ weights: [Data]) throws

    public func outputShapes(forInputShapes: [[NSNumber]]) throws -> [[NSNumber]]

    public func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws
}
```

```
public protocol MLCustomLayer {

    public init(parameters: [String : Any]) throws

    public func setWeightData(_ weights: [Data]) throws

    public func outputShapes(forInputShapes: [[NSNumber]]) throws -> [[NSNumber]]

    public func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws
}
```

```
public protocol MLCustomLayer {

    public init(parameters: [String : Any]) throws

    public func setWeightData(_ weights: [Data]) throws

    public func outputShapes(forInputShapes: [[NSNumber]]) throws -> [[NSNumber]]

    public func evaluate(inputs: [MLMultiArray], outputs: [MLMultiArray]) throws
}
```

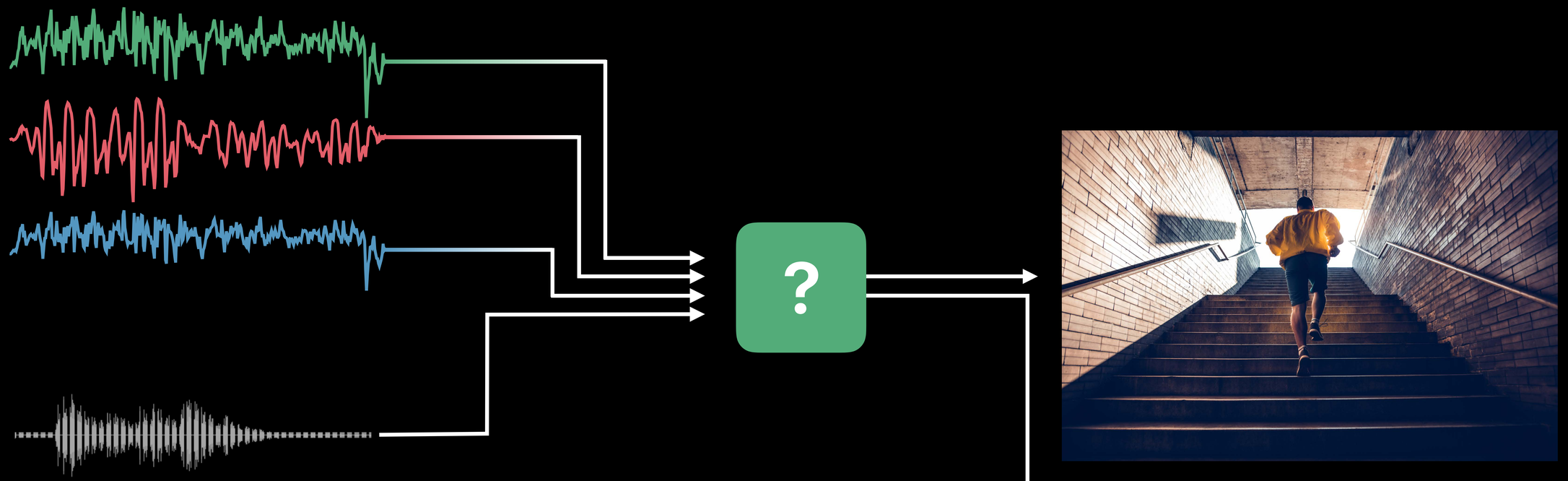
```
// Optional method for performance

optional public func encode(commandBuffer: MTLCommandBuffer,
                            inputs: [MTLTexture],
                            outputs: [MTLTexture]) throws
```

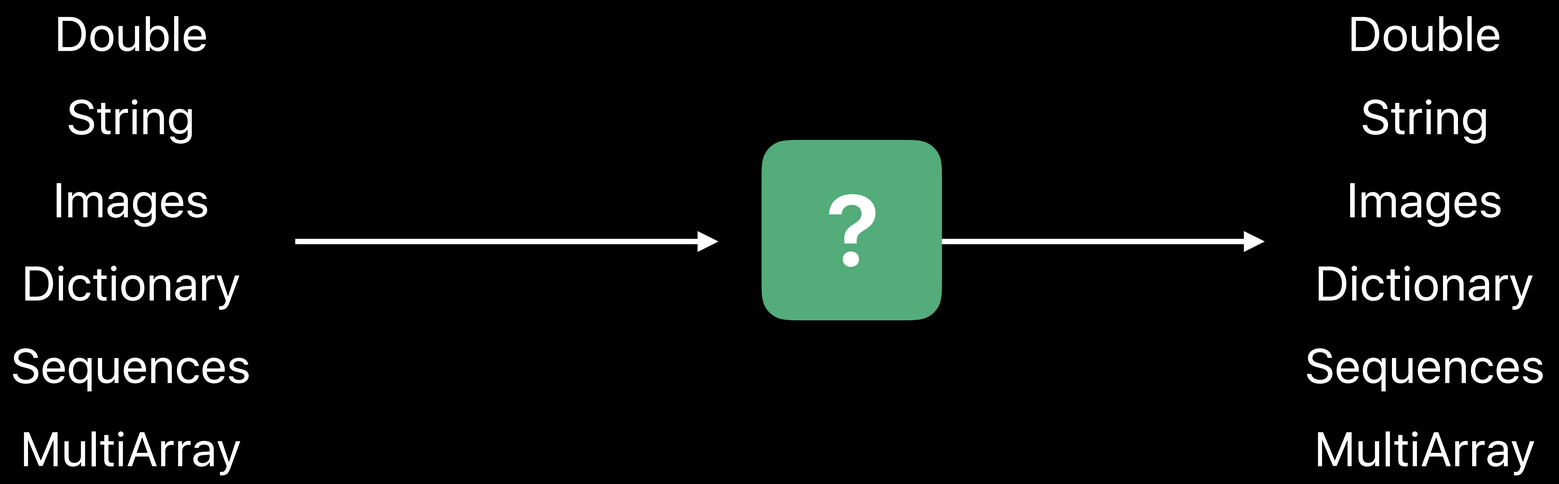
Custom Layers



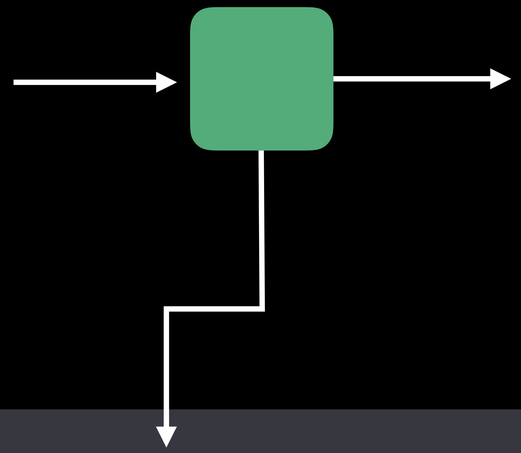

MLMODEL


?


?
“Isn’t my ring closed yet?”
“Not *yet*, but you can do it!”


Double
String
Images
Dictionary
Sequences
MultiArray
?
Double
String
Images
Dictionary
Sequences
MultiArray

Custom Models



```
@objc(AAPLCustomModel)
class AAPLCustomModel: MLCustomModel {
    . . .
}
```

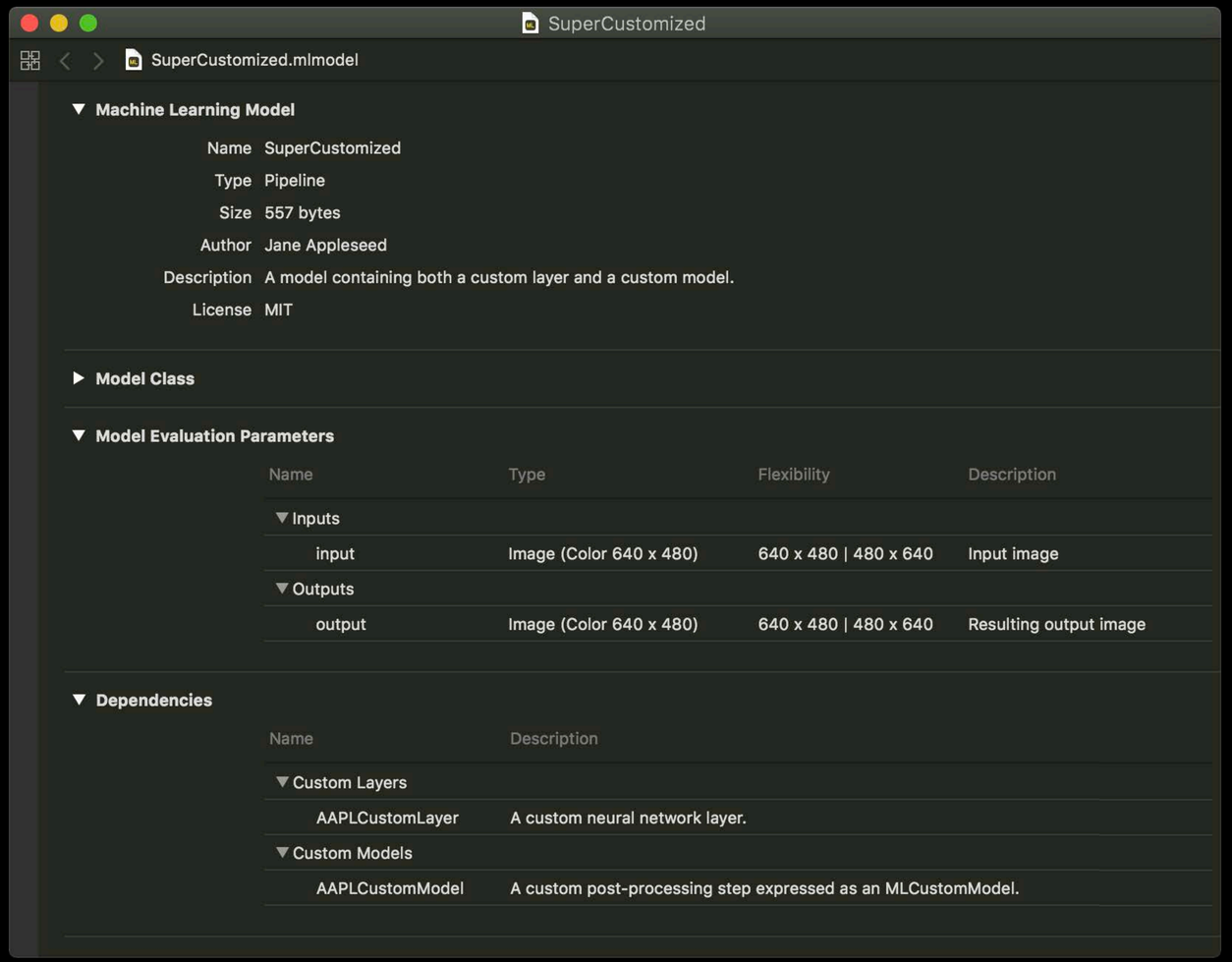
```
public protocol MLCustomModel {

    public init(modelDescription: MLModelDescription, parameters: [String : Any]) throws

    public func prediction(from: MLFeatureProvider,
                           options: MLPredictionOptions) throws -> MLFeatureProvider

    optional public func predictions(from: MLBatchProvider,
                                     options: MLPredictionOptions) throws -> MLBatchProvider

}
```

```
public protocol MLCustomModel {

    public init(modelDescription: MLModelDescription, parameters: [String : Any]) throws

    public func prediction(from: MLFeatureProvider,
                           options: MLPredictionOptions) throws -> MLFeatureProvider

    optional public func predictions(from: MLBatchProvider,
                                     options: MLPredictionOptions) throws -> MLBatchProvider

}
```

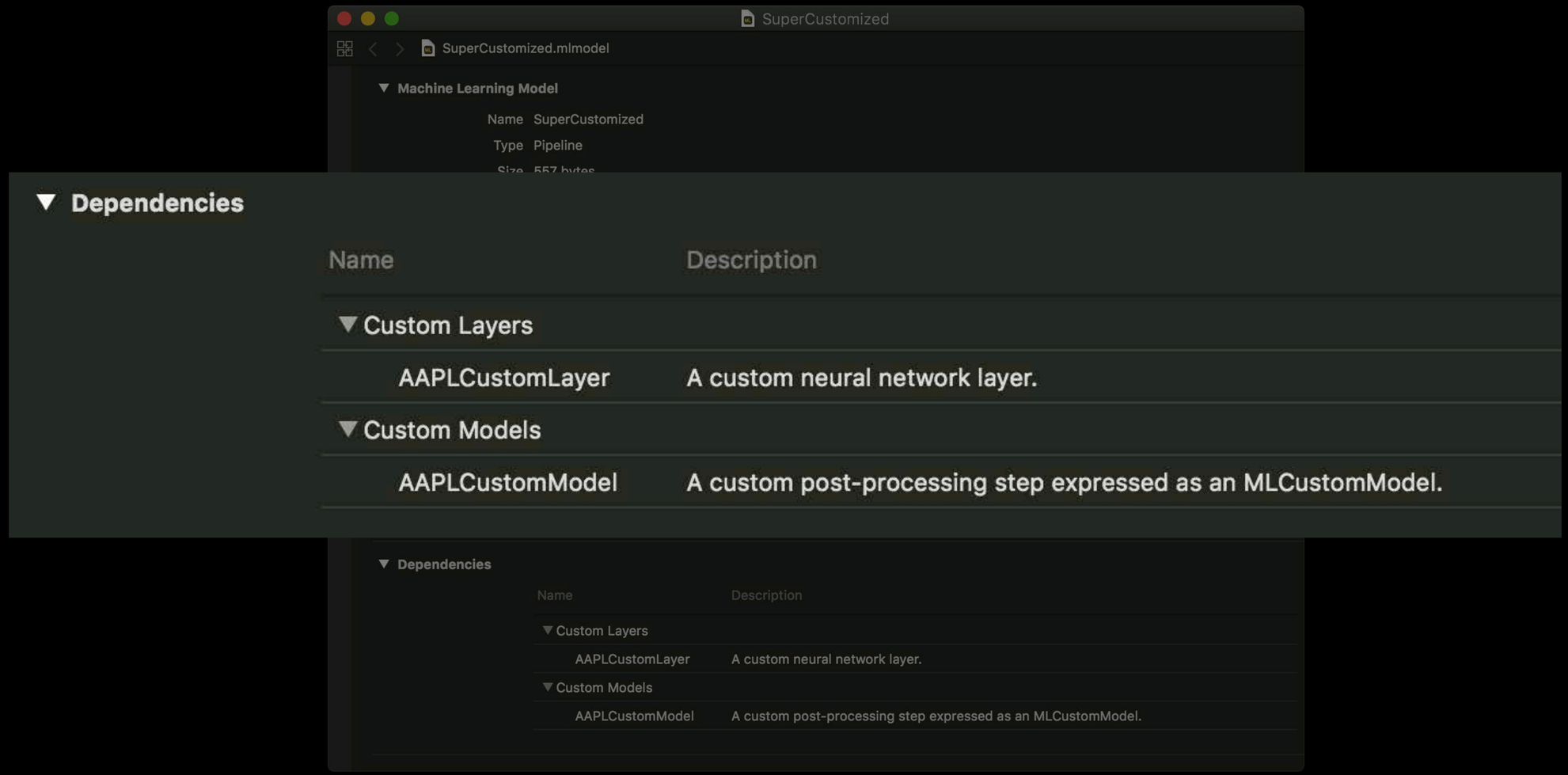
```
public protocol MLCustomModel {

    public init(modelDescription: MLModelDescription, parameters: [String : Any]) throws

    public func prediction(from: MLFeatureProvider,
                           options: MLPredictionOptions) throws -> MLFeatureProvider

    optional public func predictions(from: MLBatchProvider,
                                     options: MLPredictionOptions) throws -> MLBatchProvider

}
```

```
public protocol MLCustomModel {

    public init(modelDescription: MLModelDescription, parameters: [String : Any]) throws

    public func prediction(from: MLFeatureProvider,
                           options: MLPredictionOptions) throws -> MLFeatureProvider

    optional public func predictions(from: MLBatchProvider,
                                     options: MLPredictionOptions) throws -> MLBatchProvider

}
```

Customization in Xcode



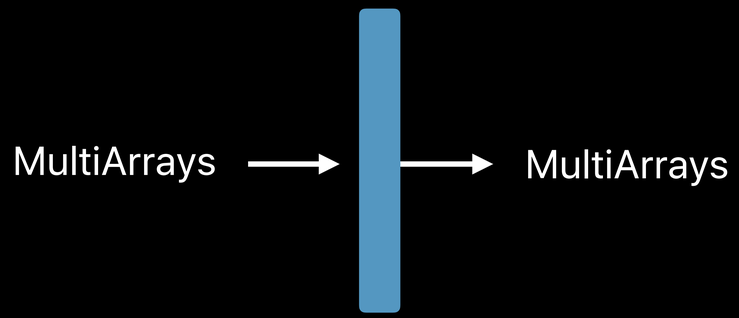
Customization in Xcode



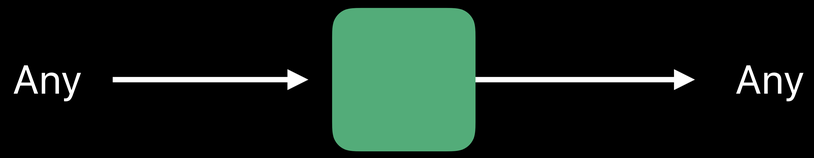
Customization

Options

Custom Layers



Custom Models



Customization

Summary

Parameters held in

Implementation provided by

+

Code in app

A Guide to Turi Create Executive Ballroom Wednesday 3:00PM

And more. . .

Non-maximum suppression | Vision feature print | Text classifiers | Word taggers | New sequence types

Core ML 2
Smaller. Faster. Customizable.

More Information

https://developer.apple.com/wwdc18/708

What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM


WWDC18