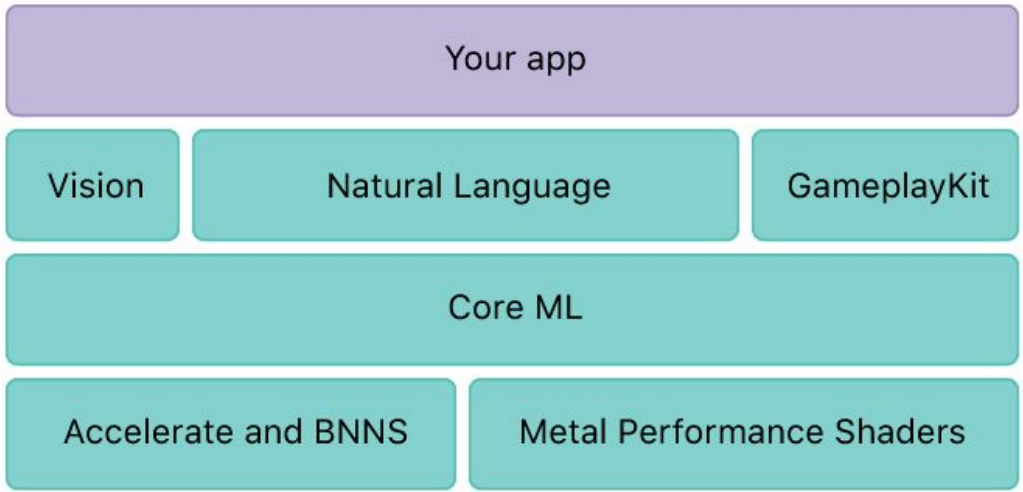
Core ML(2)

macOS Mojave required

Inhoud

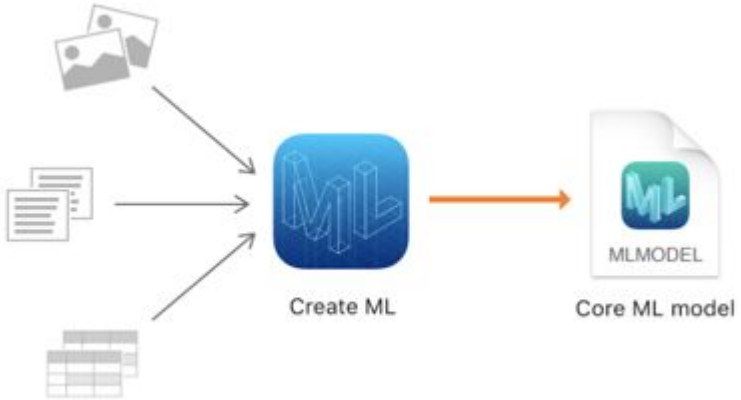
- Core ML
- Demo Cat Dog
- Demo Turi Create
- Model (and neural networks)
- Demo coremltools

Core ML

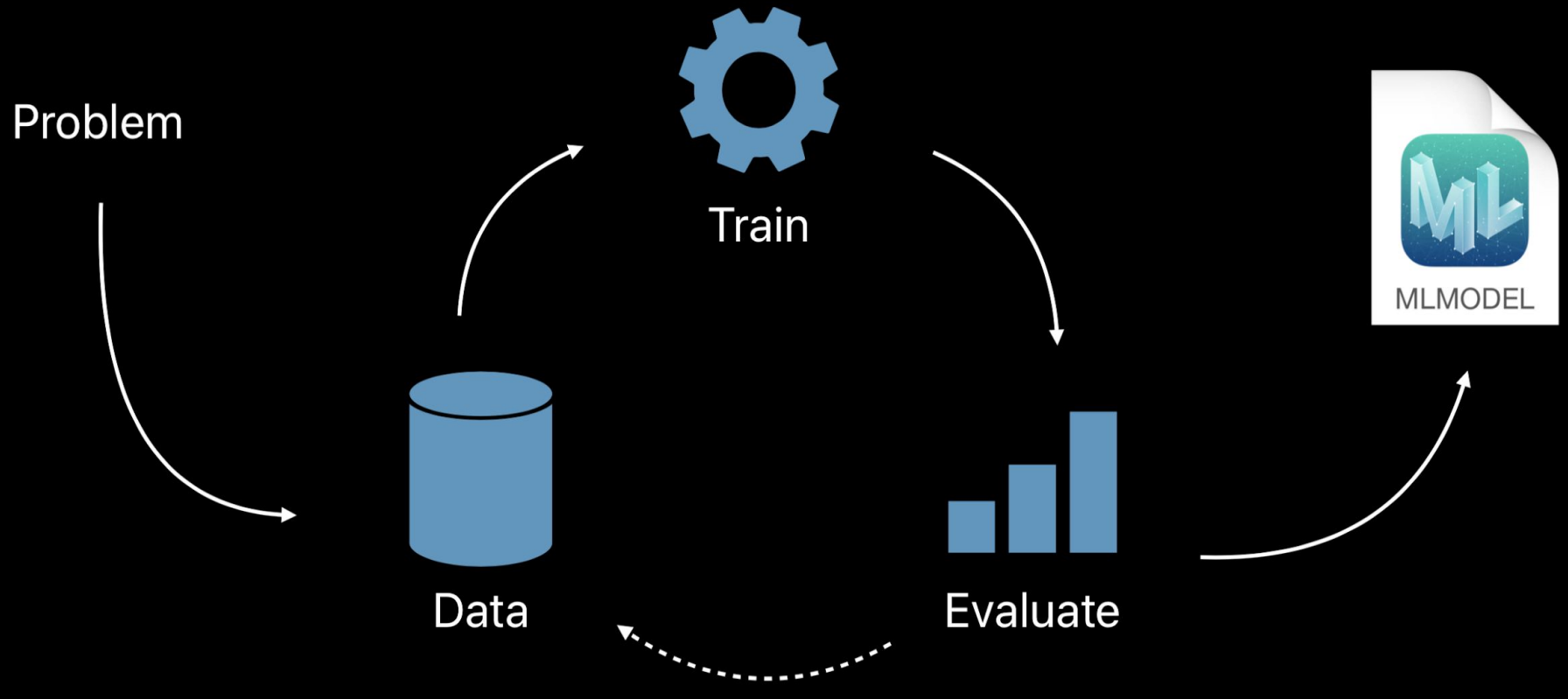

Your app
Vision
Natural Language
GameplayKit
Core ML
Accelerate and BNNS
Metal Performance Shaders

Core ML

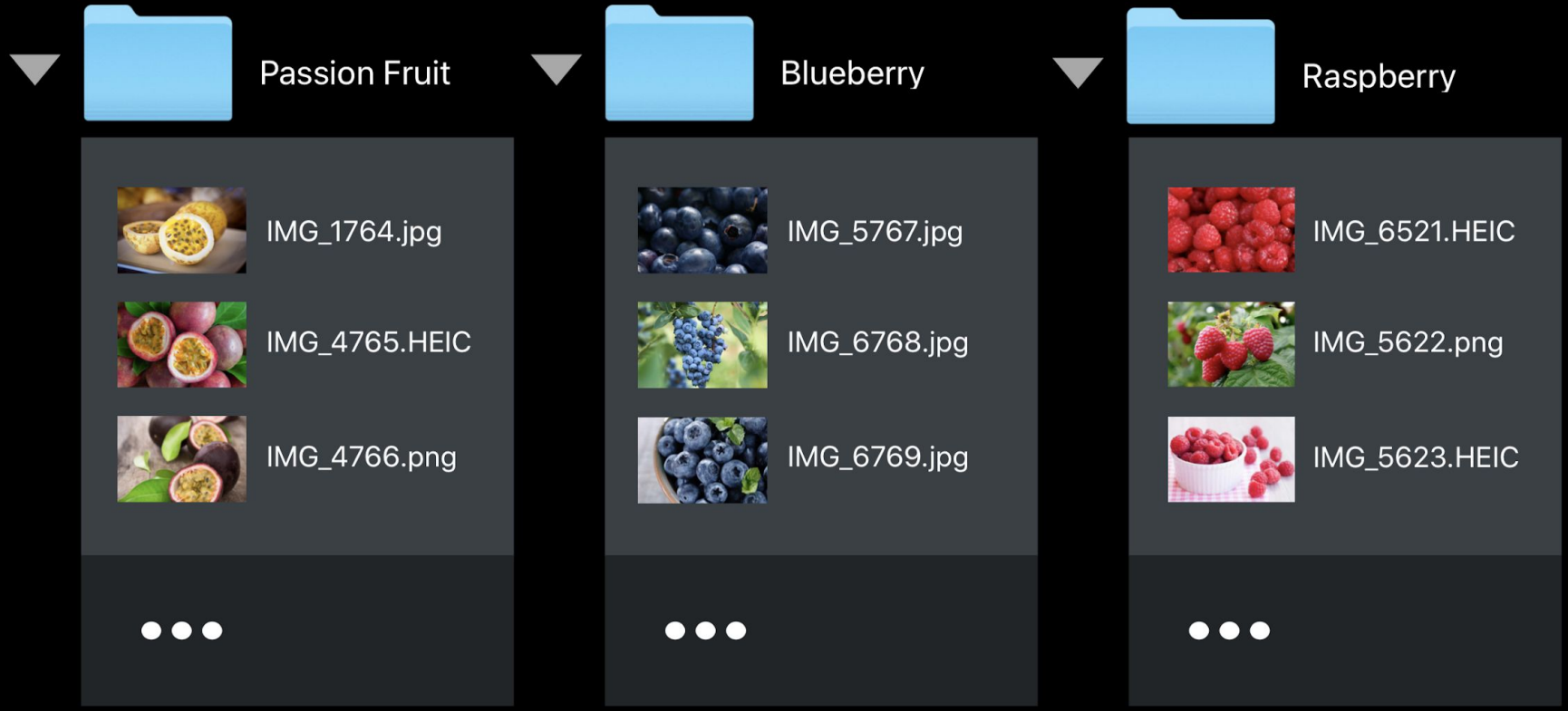
Core ML + Create ML



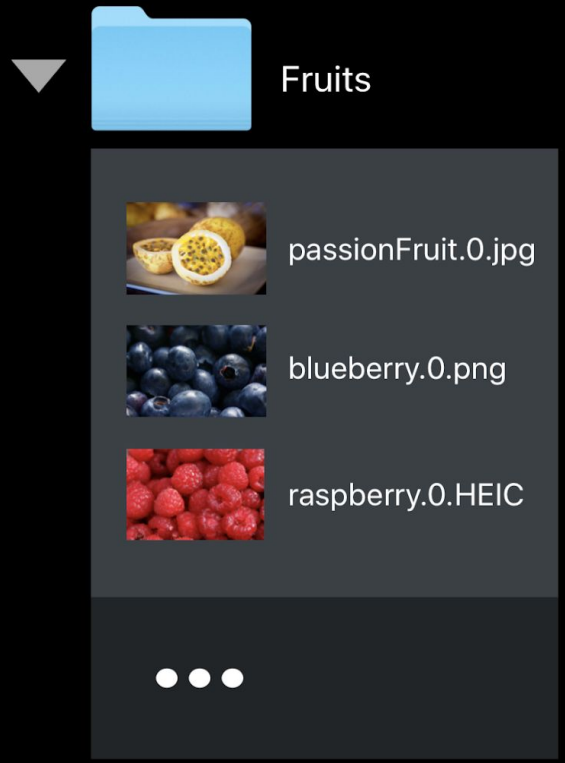
Work Flow



Data Source


Passion Fruit
IMG_1764.jpg
IMG_4765.HEIC
IMG_4766.png
Blueberry
IMG_5767.jpg
IMG_6768.jpg
IMG_6769.jpg
Raspberry
IMG_6521.HEIC
IMG_5622.png
IMG_5623.HEIC

Data Source



Requirement voor data training

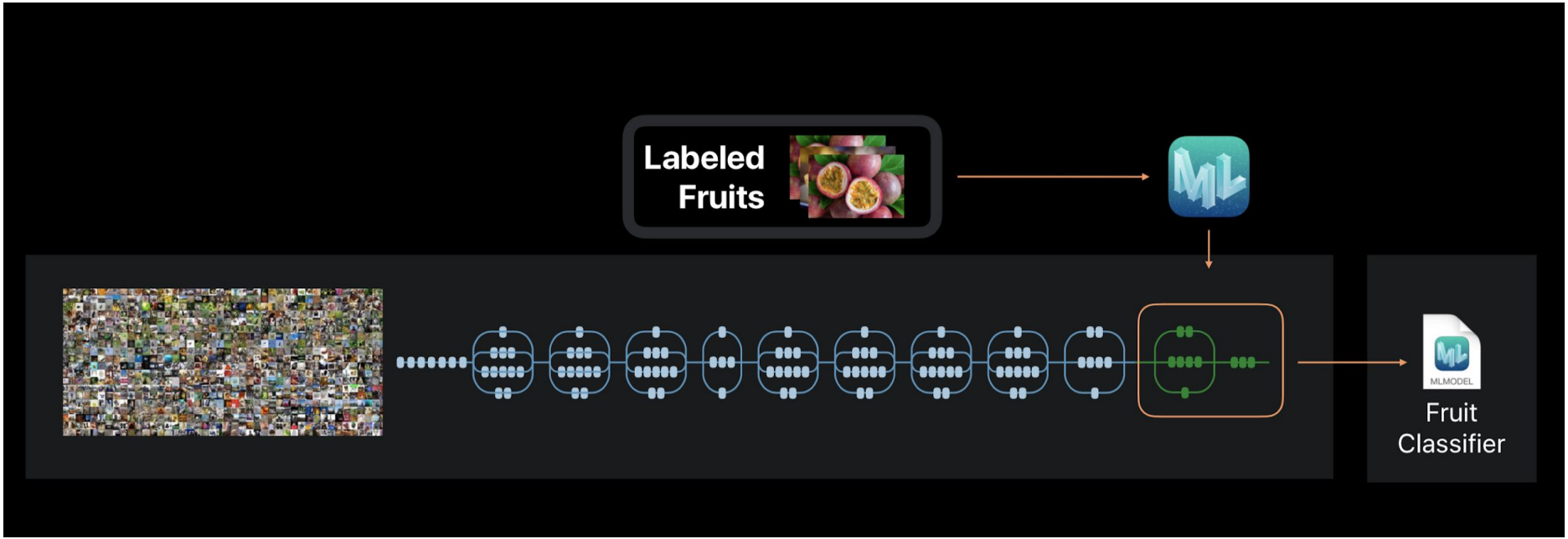
-Use at least 10 images per label for the training set, but more is always better.

-Also, balance the number of images for each label.

-It’s best to use images that are at least 299x299 pixels. (The images don’t have to be the same size as each other)

- Provide images with variety. (different angles, different conditions)

Training (transfer learning)



Opensource frameworks

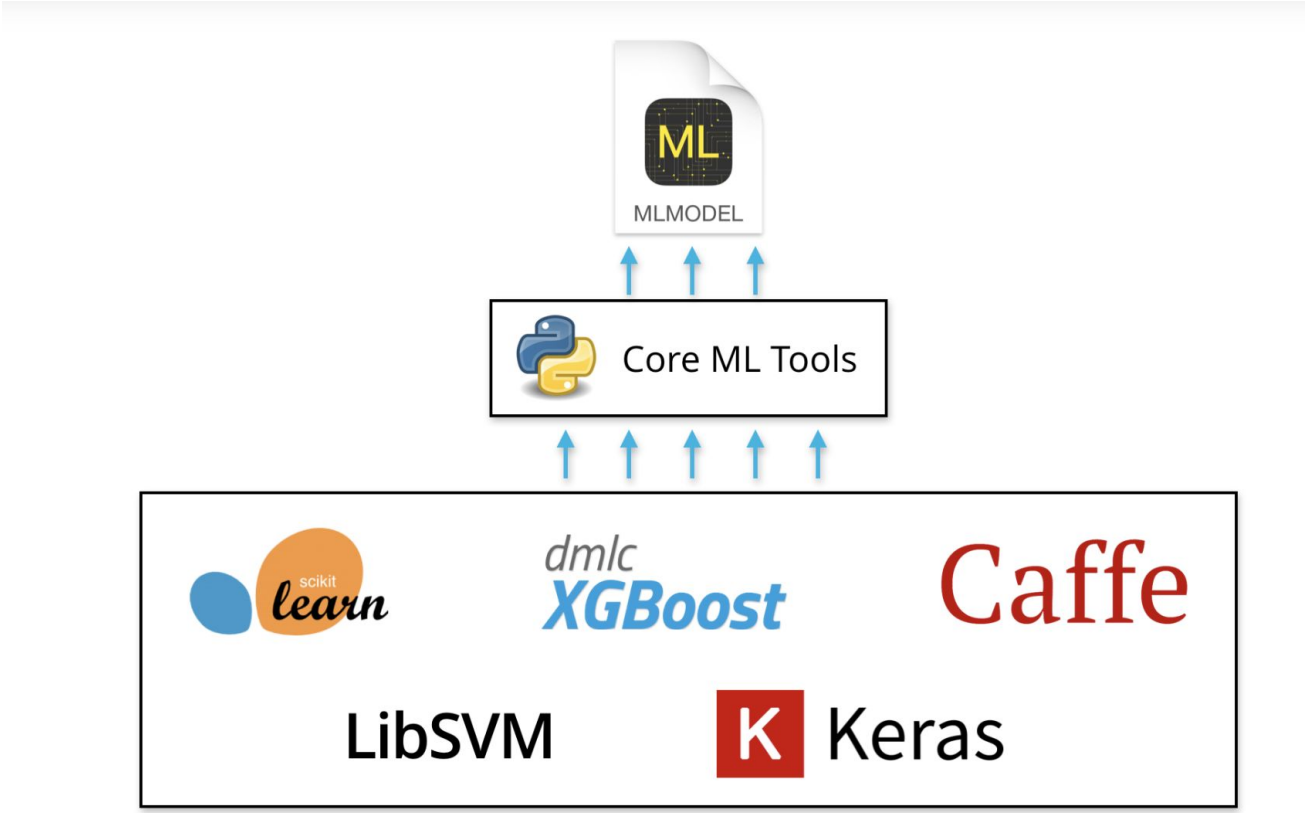


Table 1 Models and third-party frameworks supported by Core ML Tools

Model type	Supported models	Supported frameworks
Neural networks	Feedforward, convolutional, recurrent	Caffe v1 Keras 1.2.2+
Tree ensembles	Random forests, boosted trees, decision trees	scikit-learn 0.18 XGBoost 0.6
Support vector machines	Scalar regression, multiclass classification	scikit-learn 0.18 LIBSVM 3.22
Generalized linear models	Linear regression, logistic regression	scikit-learn 0.18
Feature engineering	Sparse vectorization, dense vectorization, categorical processing	scikit-learn 0.18
Pipeline models	Sequentially chained models	scikit-learn 0.18

Turi

I want to...	Machine Learning Task
Label Images	Image Classification
Recognize objects within images	Object Detection
Find similar images	Image Similarity
Create stylized avatars / profile images	Style Transfer
Personalize choices for users	Recommender
Detect an activity using sensors	Activity Classification
Analyze sentiment of messages	Text Classifier
Predict a label	Classifiers
Predict numeric values	Regression
Group similar datapoints together	Clustering

Demo Cat Dog

Wat gaan we doen?

1. Onze eigen model maken met createML
2. Dit kunnen we op 2 manieren doen (beide workflows gaan we behandelen)
3. Model gebruiken in de Camera App (Identificeren is het een kat of hond)

Images in Simulator zetten

1. cp -r imagesForTheApp/ ~/Library/Developer/CoreSimulator/Devices/BD7091AB-2BCB-44F4-8A75-D9BA8C5849BD/data/Media/DCIM/100APPLE/

Demo Turi Create

- opensource framework van apple. (Over gekocht in 2016)

Model

- Model Size
- Performance
- Customization

Dit geldt alleen voor neurale netwerken.

Model Size

- Weight Quantization = reducing the size of weight

Weight?

In case you aren't familiar with what weights are, here's a really good analogy. Say that you're going from your house to the supermarket. The first time, you may take a certain path. The second time, you'll try to find a shorter path to the supermarket, since you already know your way to the market. And the third time, you'll take an even shorter route because you have the knowledge of the previous 2 paths. Each time you go to the market, you'll keep taking a shorter path as you **learn** over time! This knowledge of knowing which route to take is known as the weights. Hence, the most accurate path, is the one with the most weights!

Quantiziation kan op diverse manieren maar coremltools gebruikt de volgende twee:

- *Linear Quantization:* is when you map the weights evenly and reduce them.

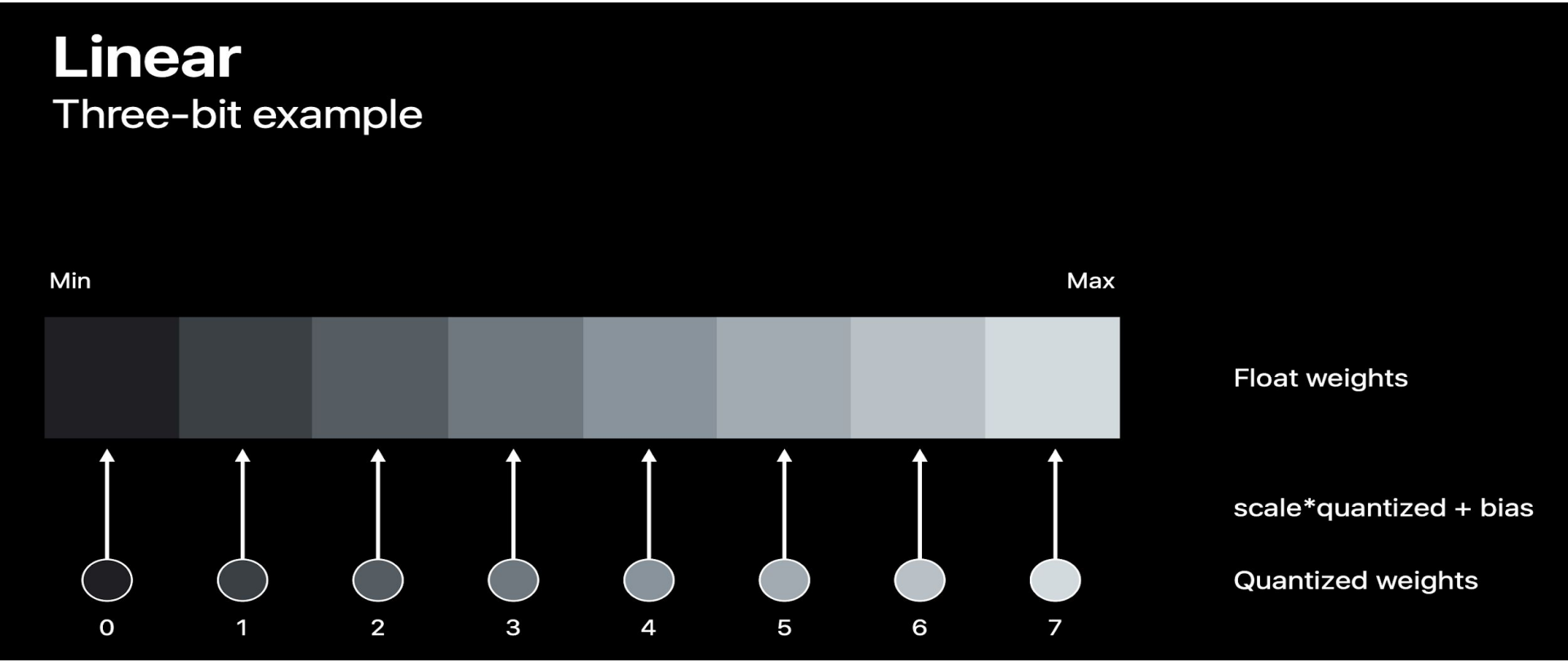
- *Lookup Table Quantization*: the model constructs a table and groups the weights around based on similarity and reduces them.

Weight Quantization

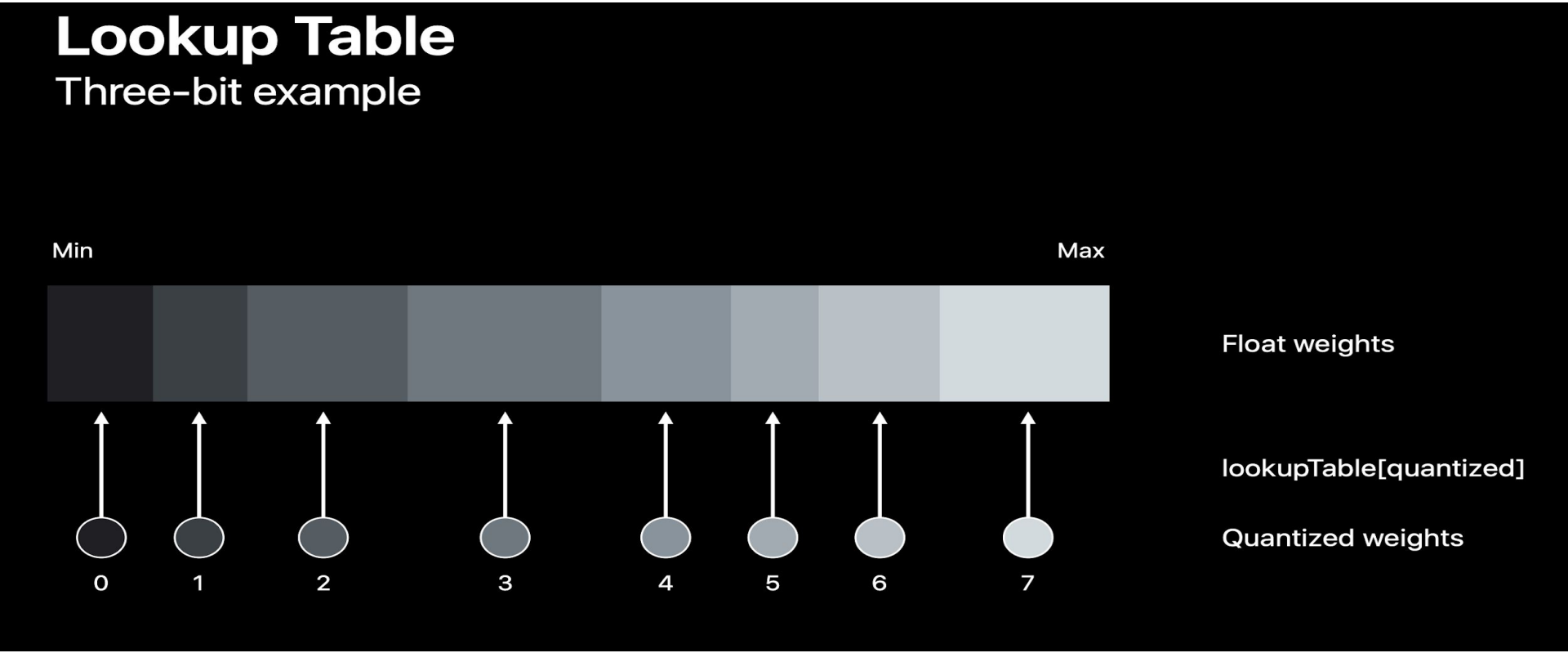
Peeking under the hood

709_Whats new in Core ML Part 2_03_Final_D

Linear Quantization



Lookup Table Quantization



Performance

Vroeger:

```
// Loop over inputs

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

Nu:

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

Customization

- Alleen voor neurale netwerken
- Van toepassing als je neurale netwerk based model van een andere framework wilt converteren naar
- MLCustomLayer (protocol)
    - Gedrag van layers kan je beinvloeden.

Before we start… with coremltools

1. python 2.7.15 installed
2. pip installed: sudo easy_install pip
3. cormltools installed: pip install coremltools

coremltools

Resources

Zie resources.txt voor gebruikte referenties.