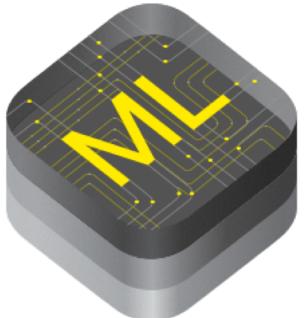


AI On Your Phone

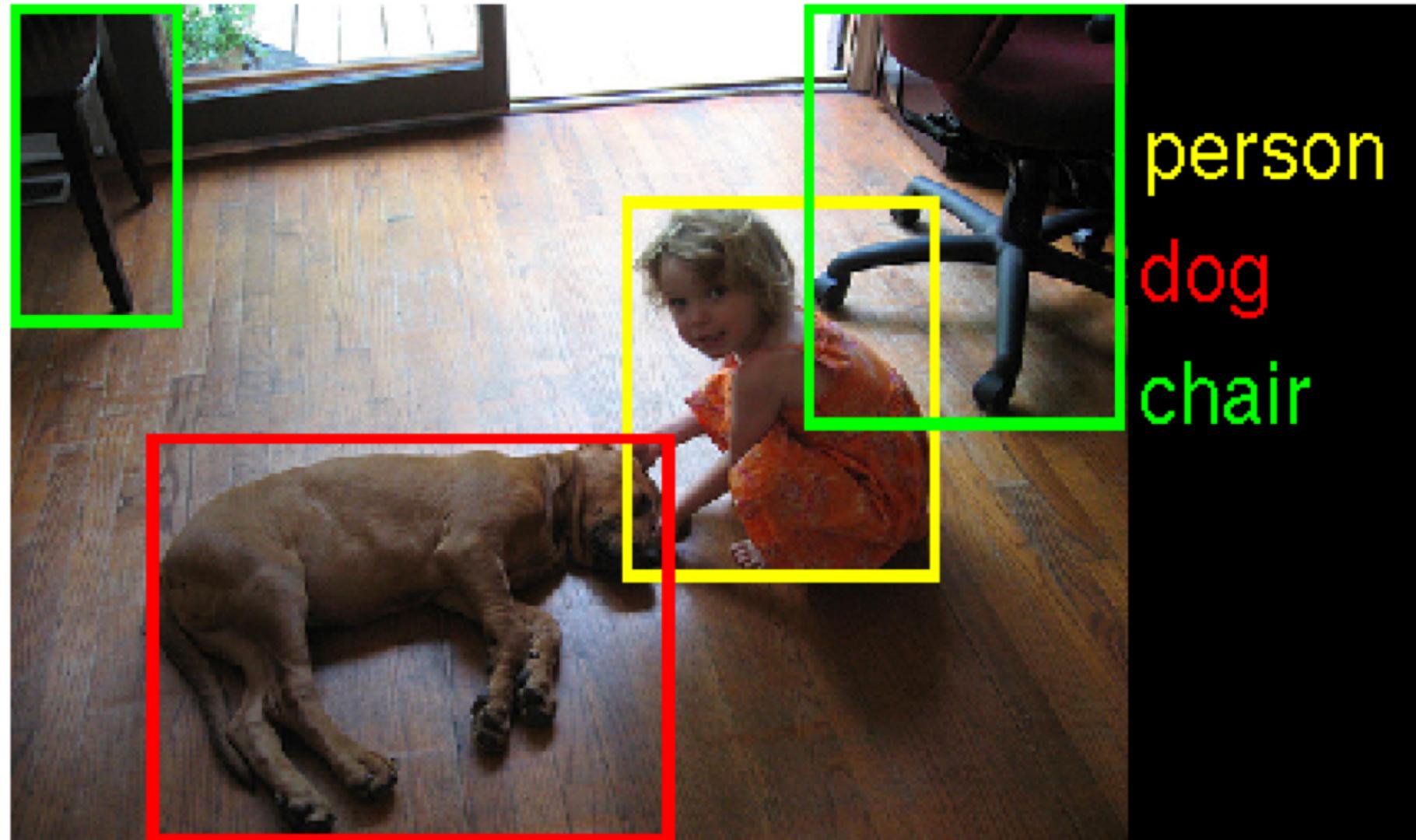
@ray_deck

github.com/rhdeck/boscc29

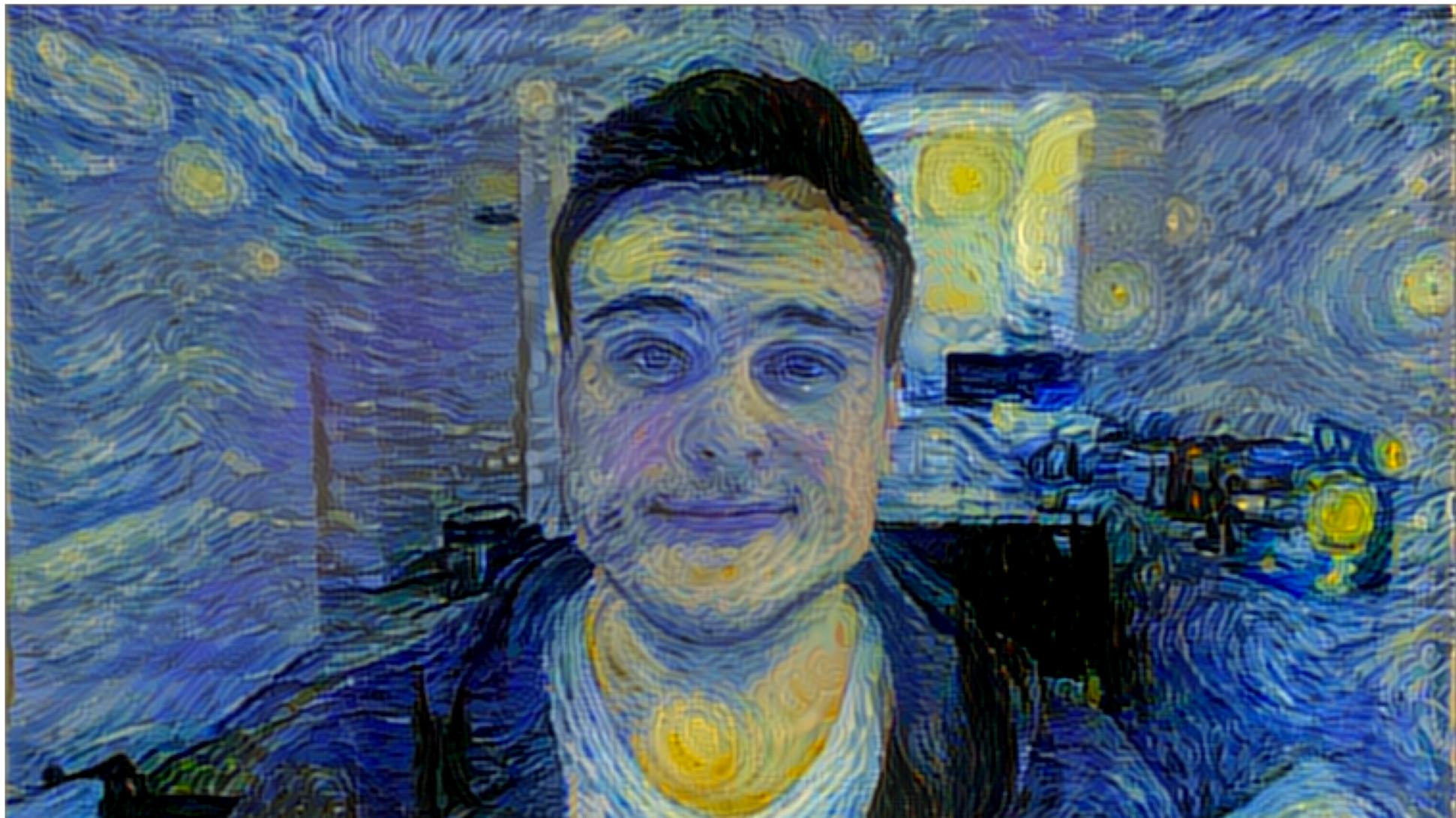


What's it Good For?

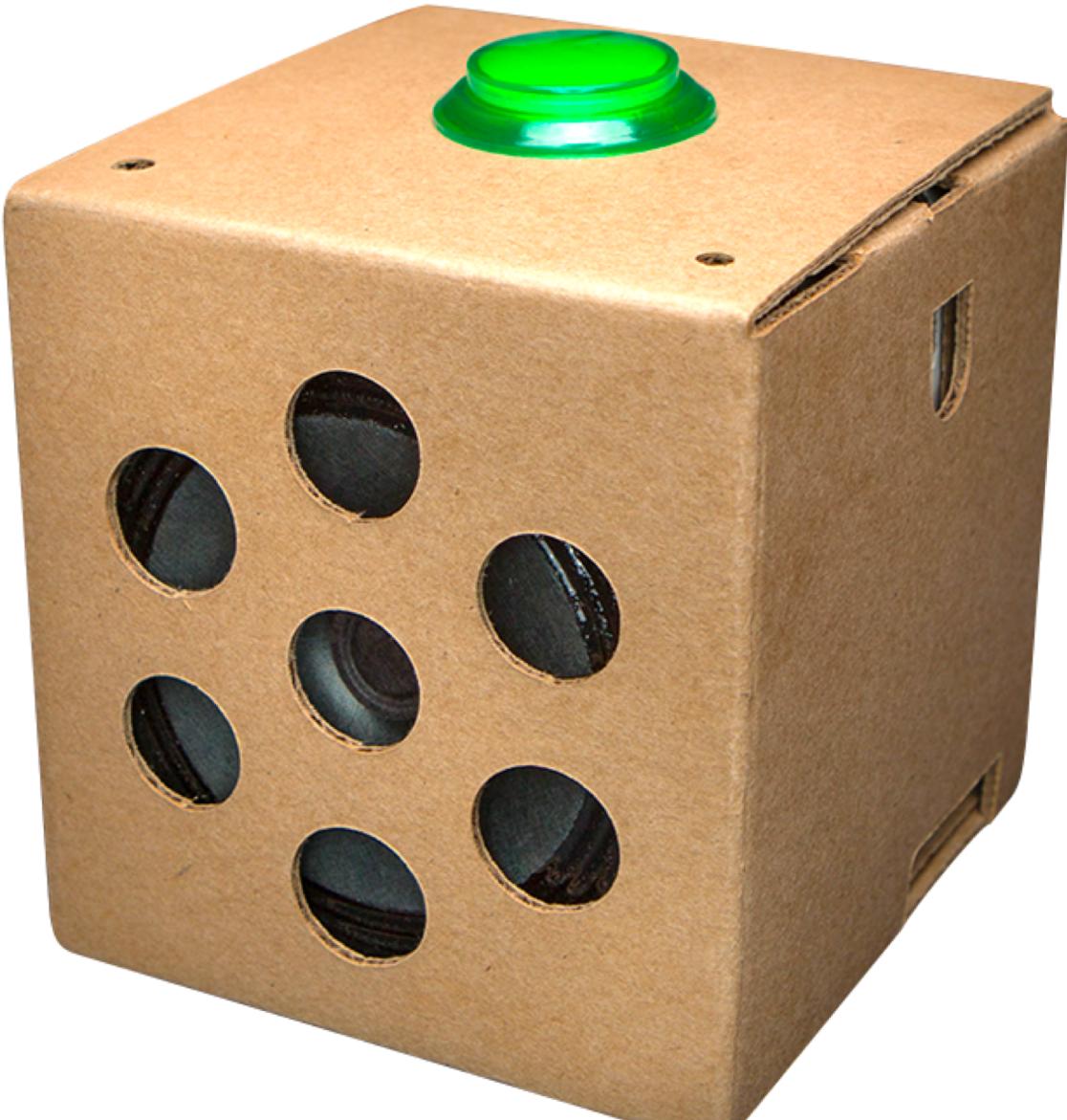
Identification



Generation



Voice



Prediction

This is a test for iOS's predictive key|



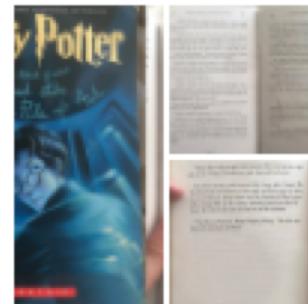
Prediction



Al Dente's Inferno
@dirtylonghair

Follow

“The Death Eaters were dead now, and Harry was hungrier than he had ever been.”



Botnik Studios @botnikstudios

We used predictive keyboards trained on all seven books to ghostwrite this spellbinding new Harry Potter chapter
botnik.org/content/harry-...

Show this thread

3:59 PM - 12 Dec 2017

What is a Machine Learning Model?

Artificial Intelligence

Deep Learning

Machine Learning

Statistical Computing

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Ops and Numbers

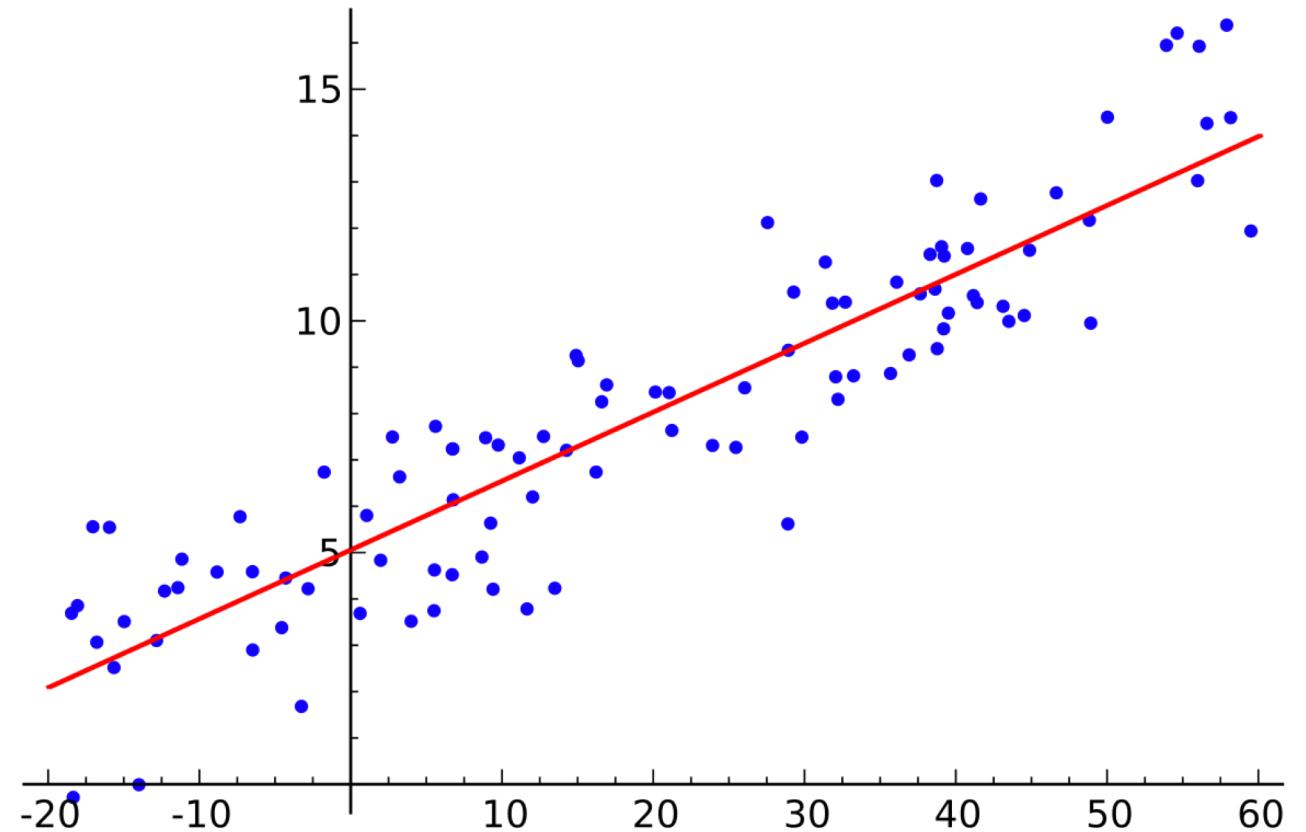
Input Shape

“The Graph”

Conv2D, ReLU, MaxPool, etc...

Output Shape

$$y = mx + b$$



Ops

Convolutions

Pooling

ReLU

TanH

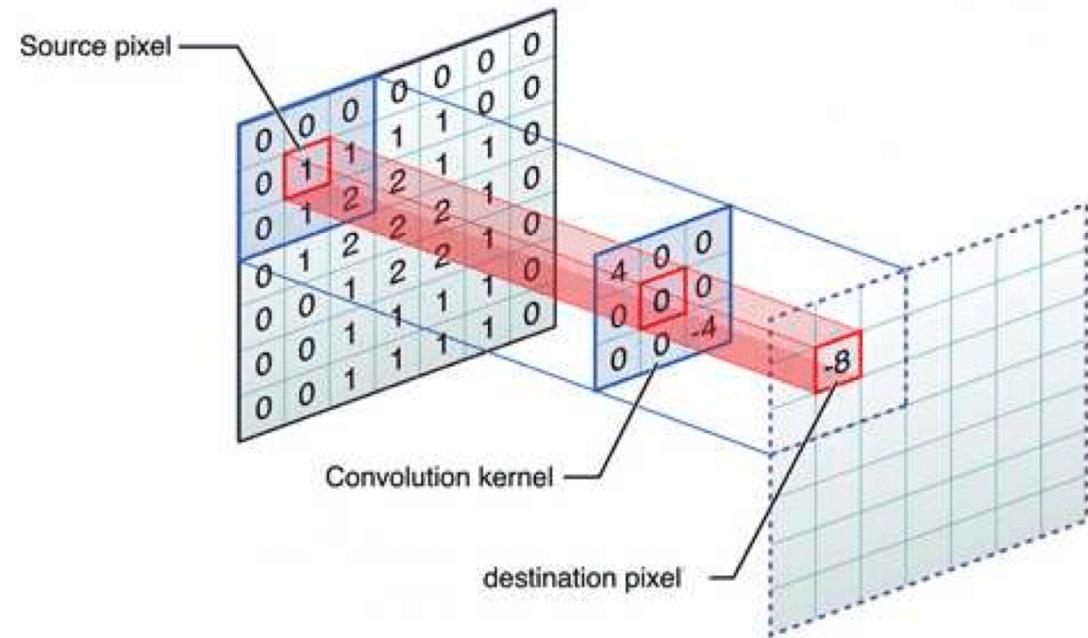
Sigmoid

LSTM

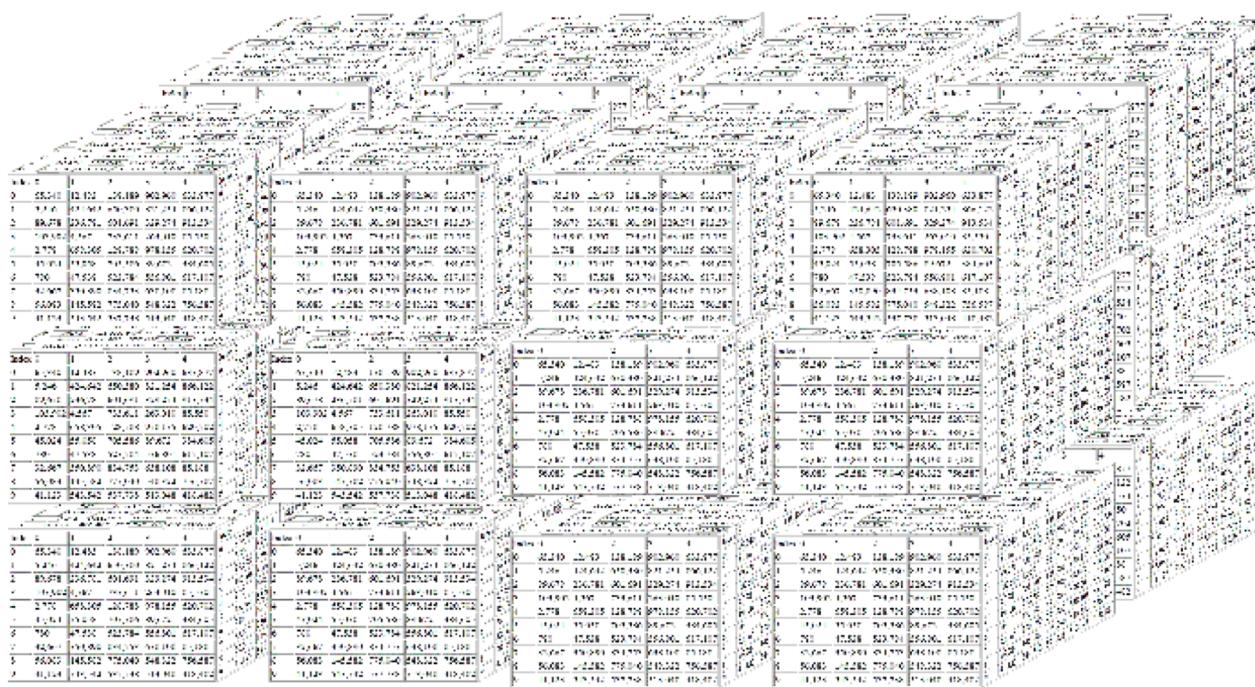
GRU

Dense/Fully-Connected

More...



High-Dimensional Numbers



Images are High-Dimensional Numbers!

Array RGB						
Page 3 – blue intensity values	0.689	0.706	0.118	0.884
Page 3 – blue intensity values	0.535	0.532	0.653	0.925
Page 3 – blue intensity values	0.314	0.265	0.159	0.101
Page 3 – blue intensity values	0.553	0.633	0.528	0.493
Page 3 – blue intensity values	0.441	0.465	0.512	0.512
Page 2 – green intensity values	0.342	0.647	0.515	0.816	...	0.421
Page 2 – green intensity values	0.111	0.300	0.205	0.526	...	0.912
Page 2 – green intensity values	0.523	0.428	0.712	0.929	...	0.219
Page 2 – green intensity values	0.214	0.604	0.918	0.344	...	0.128
Page 2 – green intensity values	0.100	0.121	0.113	0.126	...	0.133
Page 1 – red intensity	0.112	0.986	0.234	0.432	...	0.204
Page 1 – red intensity	0.765	0.128	0.863	0.521	...	0.760
Page 1 – red intensity	1.000	0.985	0.761	0.698	...	0.997
Page 1 – red intensity	0.455	0.783	0.224	0.395	...	0.995
Page 1 – red intensity	0.021	0.500	0.311	0.123	...	0.726
Page 1 – red intensity	1.000	1.000	0.867	0.051
Page 1 – red intensity	1.000	0.945	0.998	0.893
Page 1 – red intensity	0.990	0.941	1.000	0.876
Page 1 – red intensity	0.902	0.867	0.834	0.798

Old Stuff Works

<https://arxiv.org/pdf/1803.01271.pdf>

An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling

Shaojie Bai¹ J. Zico Kolter² Vladlen Koltun³

Abstract

For most deep learning practitioners, sequence modeling is synonymous with recurrent networks. Yet recent results indicate that convolutional ar-

chine translation (van den Oord et al., 2016; Kalchbrenner et al., 2016; Dauphin et al., 2017; Gehring et al., 2017a;b). This raises the question of whether these successes of convolutional sequence modeling are confined to specific applications, or if they can be generalized to a wider range of

A photograph of a standard cardboard egg tray containing a dozen brown eggs. The tray is open at the top, showing two circular holes for hanging. The eggs are nestled in grey egg carton inserts. The word "Packaging" is overlaid in a large, black, sans-serif font across the center of the tray.

Packaging

Models (Can Be) Files

Shapes of Input(s)

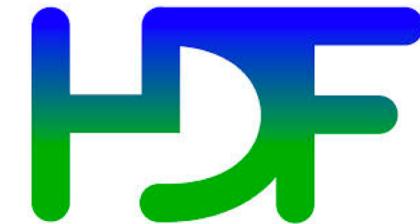
Operations

Hyperparameters

Weights

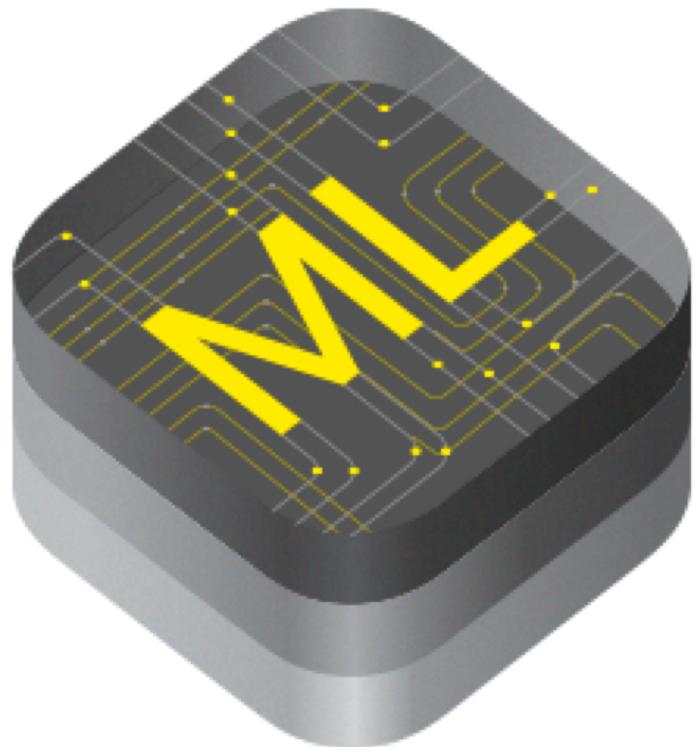
Sources & Drains

Shapes of Output(s)



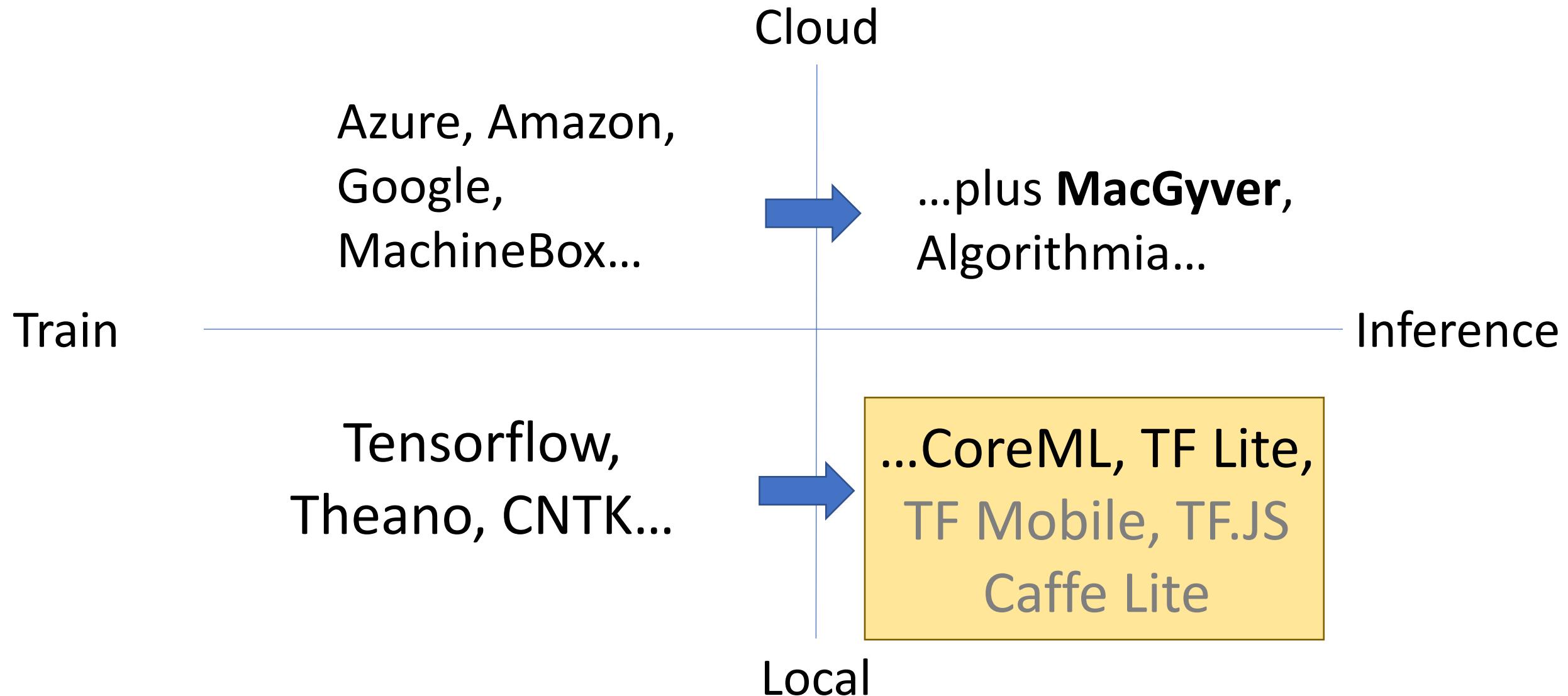
What Do we Need?

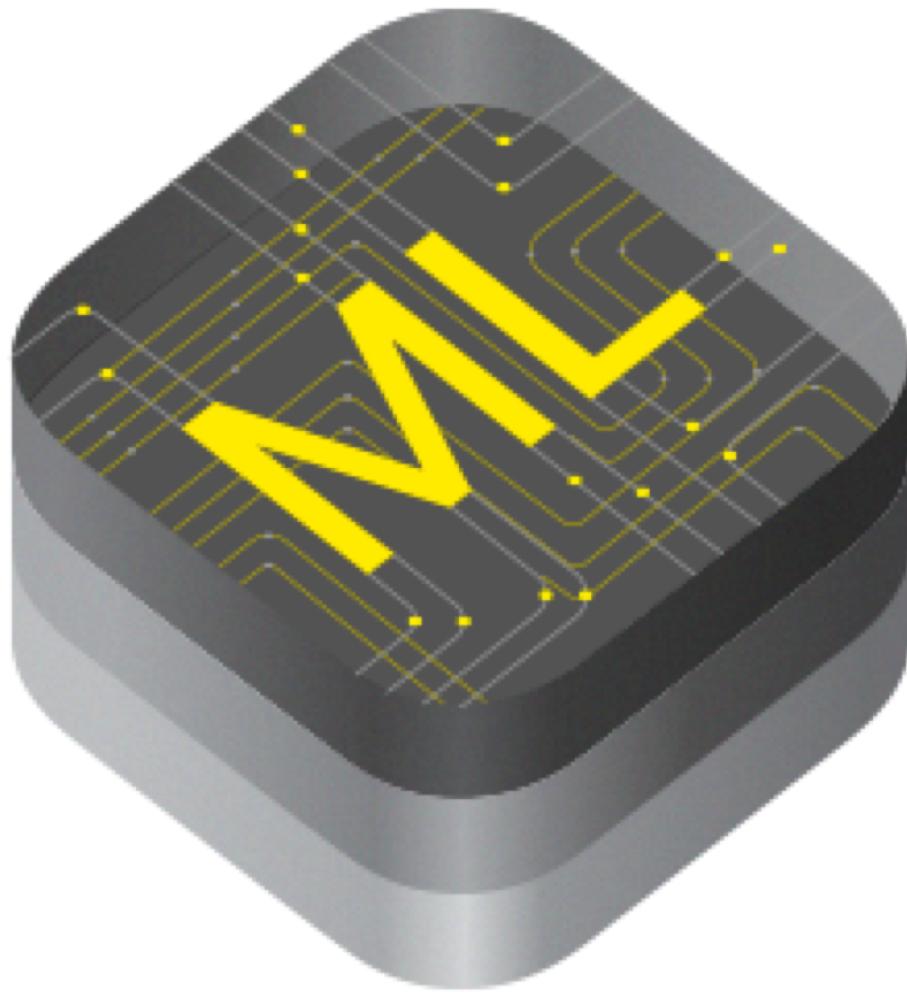
- 1. Convert Image/Frame to Normalized Number**
- 2. Run the model efficiently on local resources**
- 3. Convert output number to what we seek:**
 - Dictionaries of labels
 - Generated Image
 - Prediction
 - etc

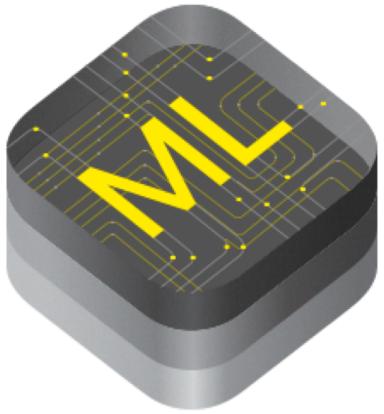


TensorFlow
Lite

ML Universe





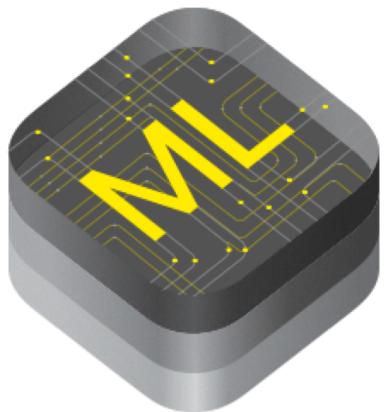


iOS Runtime and Bindings

File Format

Conversion Tooling

Runtime and Bindings



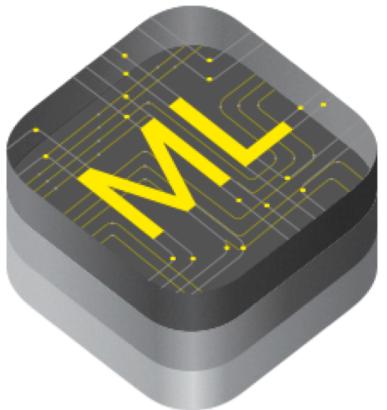
CVPixelBuffer in, Dictionary Out

Support for OTA/Runtime
Download and Compilation

Uses Metal Shaders for processing

Limited Operation Support

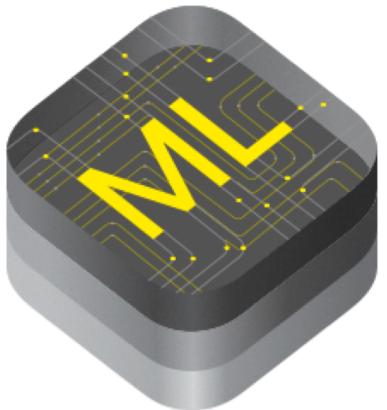
.mlmodel



1. Protobuf
2. Feature Normalization
 - Value Range of Numbers (0-255 -> 0-1)
 - Mean Value
3. Channel Ordering
 - BGR vs RGB
4. Output Labeling/Denormalization

Conversion

Keras – coremlconverter



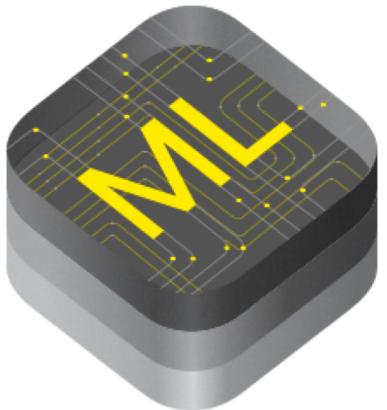
Tensorflow – tf-coreml

CNTK - mmdnn

MXNet – mxnet-to-coreml

Advantages

Ease of Application



Speed

Multiple analyses per second

Separation of Concerns

Compilation vs Inference

Source Agnostic





Lite

Multiple Runtimes
File Format
Converter

Cross-Platform



Lite

iOS, Android, Raspberry Pi

feed() and run()

No Compilation Step

Limited Operations

Number-to-Number

.tflite



Lite

FlatBuf

Quicker Memory Access

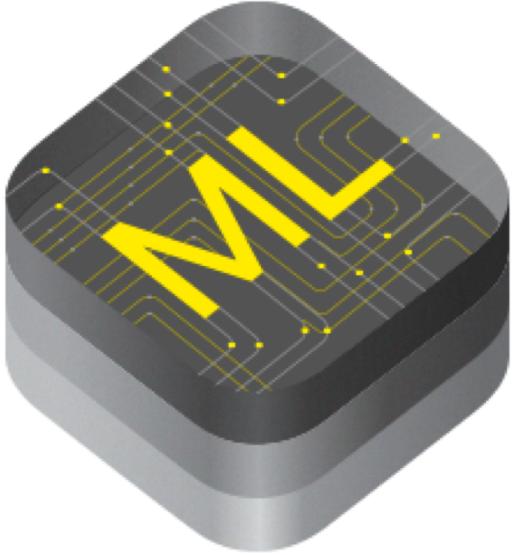
Alternative Precision: Bfloat16

Conversion



Lite

toco
Tensorflow Optimizing Converter

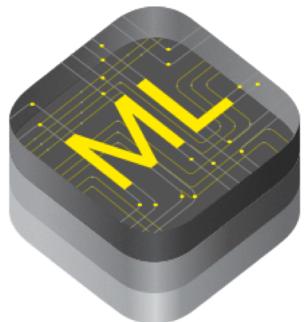


iOS Only
Input/Output Affordance
Production
Chunky

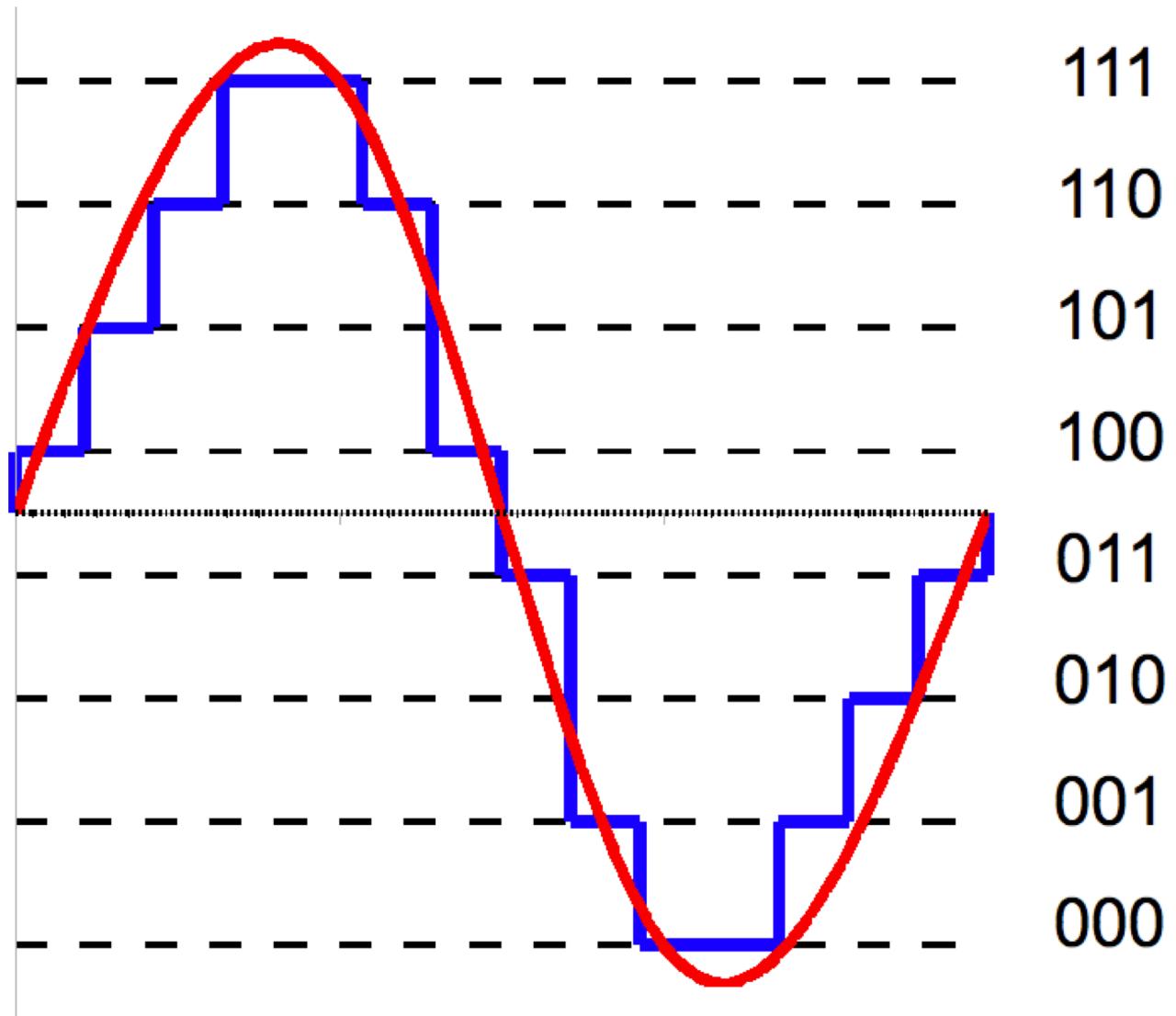


Multiplatform
Just the Numbers
Beta
Small

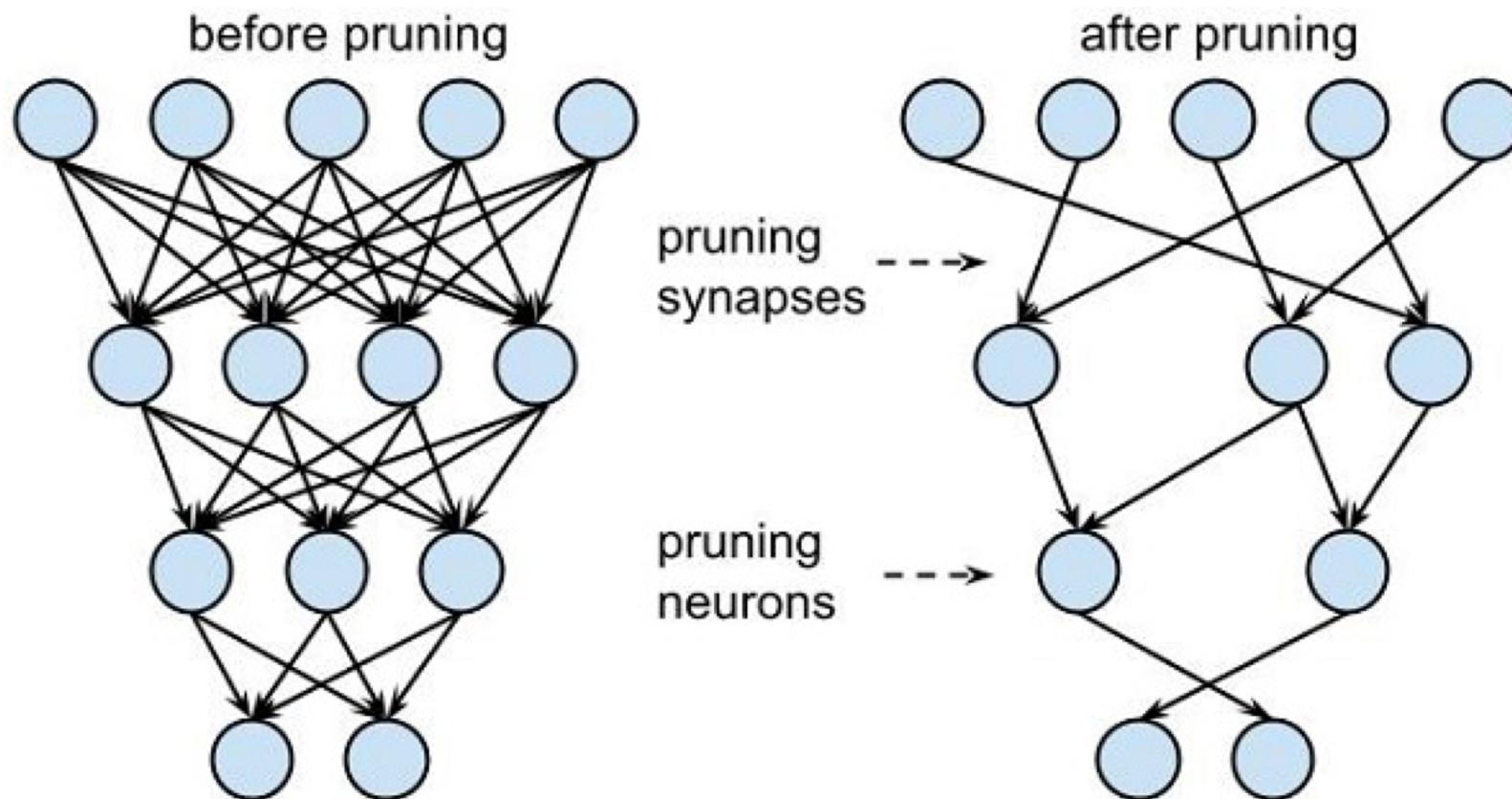
Cross-Platform Optimizations



Quantization



Pruning



Planning

VGG16

550MB

InceptionV3

94MB

MobileNet

17MB

One More Thing





WebGL

Functional Programming

Cross-Platform Nirvana?



@ray_deck
github.com/rhdeck/boscc29