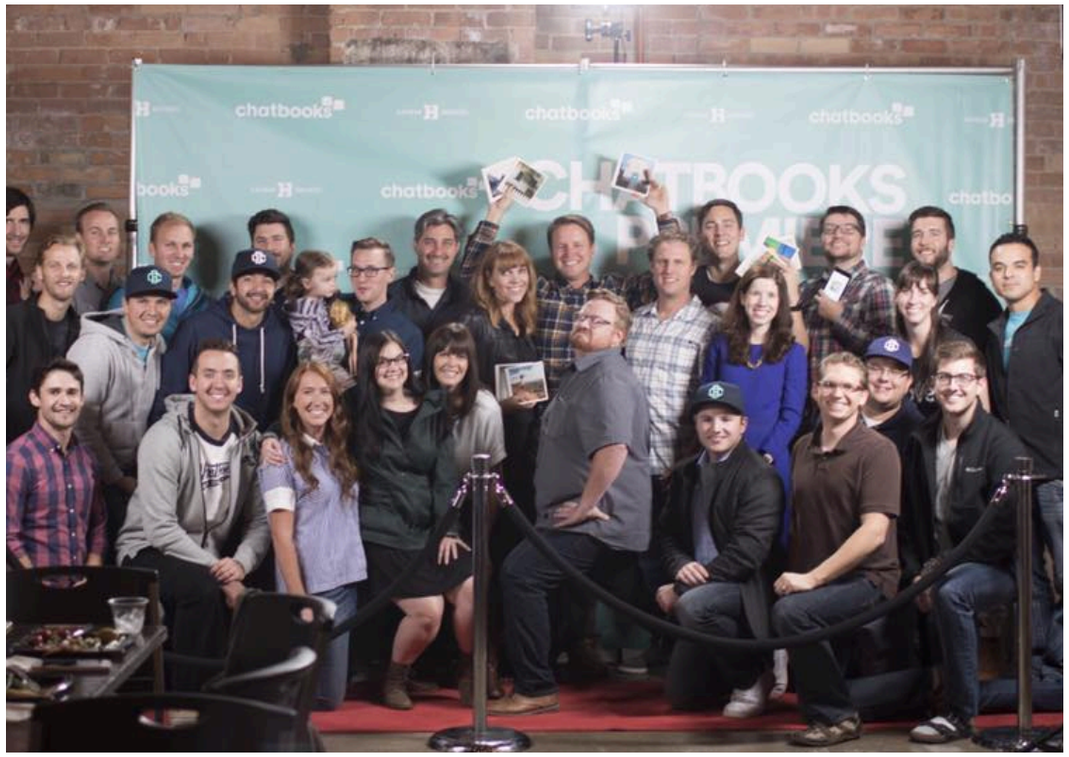
How to Port, Deploy, & Personalize a Deep Neural Network on iOS

Utah Deep Learning Meetup
October 17, 2017

Jared Heywood

Chatbooks - Future Glassblower

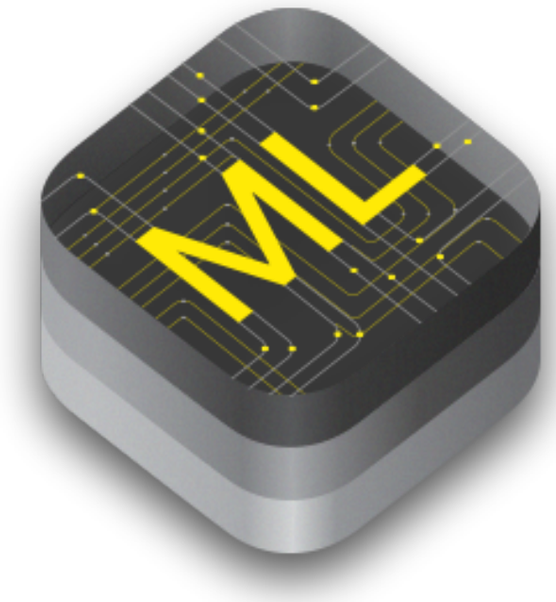


Acknowledgements

[Github - Repo]

[Stack Overflow Answer]

1. Intro to Core ML
2. Port Model to Core ML
3. Deploy on iOS
4. Update Models
5. Demo

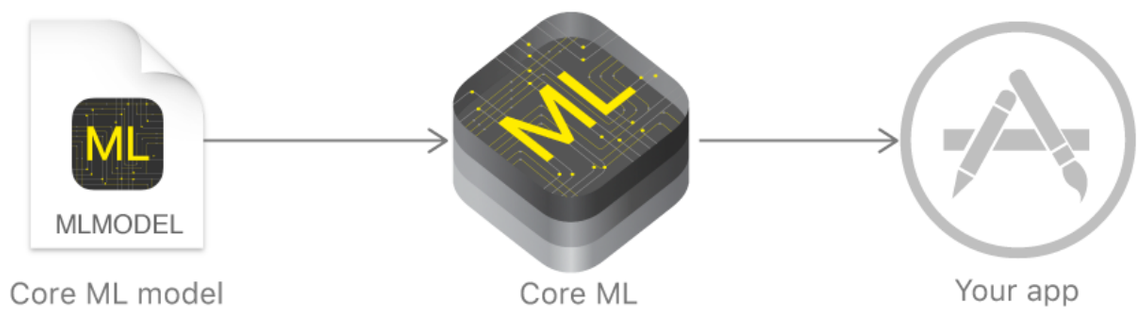


- Access Apple’s Vision Outputs
- Deploy your own DNN’s
- Optimized for accelerated inference

Porting A Trained Model

Nuances to be aware of:

- Version Nightmare
- Image Scaling
- RGB Bias
- Feature Labeling



```
coreml_model = coremltools.converters.keras.convert(model=model,
            input_names='image', output_names='probabilities',
            image_scale=2./255, red_bias=-1, green_bias=-1, blue_bias=-1,
            class_labels='./imagenet_labels.txt')

coreml_model.save('./DemoModel.mlmodel')
```

Deploy the Model

It really can be as simple as:

```
let model = MyTrainedModel()
let prediction = model.prediction(image: pixelBuffer!)
```

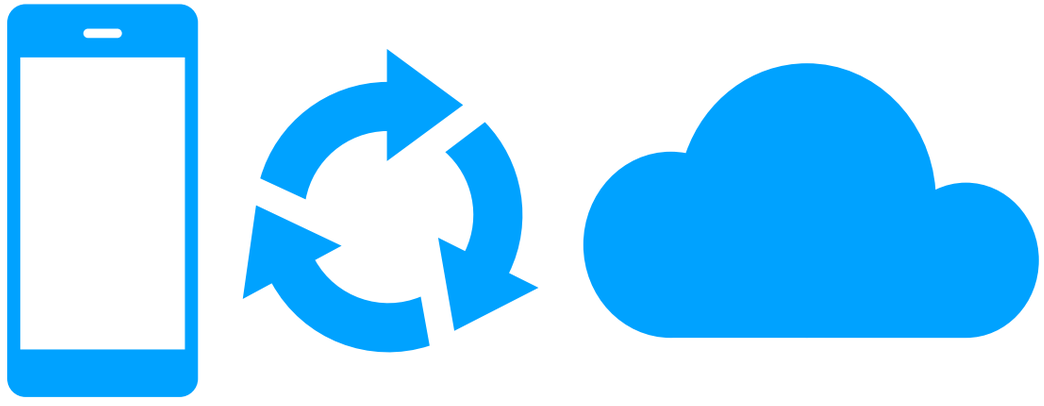
But… unfortunately that is the worst way

```
import Vision
import CoreML

let model = try VNCoreMLModel(for: MyTrainedModel().model)
let request = VNCoreMLRequest(model: model,
completionHandler: myResultsMethod)
let handler = VNImageRequestHandler(url: myImageURL)
handler.perform([request])
```

```
func myResultsMethod(request: VNRequest, error: Error?) {
    guard let results = request.results as?
[VNClassificationObservation]
        else { fatalError("huh") }
    for classification in results {
        print(classification.identifier, // the scene label
            classification.confidence)
    }

}
```

Update the Model

1. User makes decision
2. Data securely sent to cloud
3. Fine tune model
4. Deploy new version



	Dynamic parts	Size	Initial loading time
Compiled and bundled with the app	Nothing	Large	None
MLModel.compileModel()	Everything	Smaller than compiled	Short
Manual compression, updating weights	Weights	Smallest	Long
Manual compression, shipping a new model	Weights, network	Smallest	Long

Thanks to [Camilla Dahlstrom]

DEMO

Potential Impact

- Saved Inference Cost
- Saved Time on Server Communication
- New Businesses?
- Distributed Computing