Machine Learning for iOS

Machine learning has been subject to great hype within the technology industry; sparking great interest amongst seasoned and aspiring engineers alike. While the technology has brought great promises to the kinds of problems it can solve, the steep learning curve of the core functionality of machine learning would deter even the most seasoned engineers. Despite the learning curve, high level libraries the likes of Keras, SKLearn, and Tensorflow help democratize machine learning to the masses. Apple made their contributions by developing Core ML and Turi Create; high level frameworks for model creation and on-device inference. With it, developers can automate the detection of explicit images using image recognition or create chatbots for help centers with sentiment analysis.

Hello, Machine Learning

Machine learning is a topic in computer science where computers can make inferences without being explicitly instructed on what to infer. A common machine learning application would be image recognition; where the device can accurately recognize the dominant object in an image.

From an eagles eye view, the idea of machine learning is fairly simple. Based on data, you are modeling the unique features that describe the data. As an example, when creating an image recognition app that can classify dogs and cats, and machine learning algorithm will sort through images of dogs and cats and parse unique features that would represent a dog and a cat separately. In the end, you end up with a model that describe the unique features of a dog and a cat.

Based on this analogy, you can foresee some of the challenges with machine learning. One of which is how do you create these models? That is a topic for another day. Another issue is where do you put that model? and lastly, how can you improve on an existing model.

Thanks to the smart people from the Googles and Apples of the world, we have high level libraries that ease the development of these models along with some services on how to host these models.

What is Core ML

Straight from the Core ML documentation, Core ML is a machine learning framework for on-device inferences. Therefor, it is not the type of machine learning library the likes of TensorFlow or Keras. You can think of it as a formatted model for iOS. Since it is designed by Apple, you can be assured that these models are optimized to run on-device. It can take advantage of the CPU, GPU, or Metal; depending on the available hardware. One of the advantages of Core ML is that it can also be paired with other Apple ML libraries like Vision, NaturalLanguage, and GameplayKit.

How to use Core ML

The use of these models will depend on the application. While you can directly use the model for inferences, some applications may work better when paired with one of Apple's ML libraries.

Here is an example of how to make image inferences with Core ML and the Vision framework

```
func classify(image: UIImage) {
    let model = try! VNCoreMLModel(for: Pets().model)

    let visionRequest = VNCoreMLRequest(model: model) { (request, error) in

        let results = request.results as! [VNClassificationObservation]

        let identifier = results.first?.identifier
        let confidenceLevel = results.first?.confidence

        print(identifier!, confidenceLevel!)

    }

    try! VNImageRequestHandler(cgImage: image.cgImage!, options: [:]).perform([visionRequest])
}
```

How to create Core ML Models

The easiest way to develop these models is to use a python library called Turi Create. Straight from the documentation, it streamlines the development process of core ml models without having to know anything about machine learning.

Here is an example of how to create and image classifier with TuriCreate

```
import turicreate as tc

data = tc.load_images('image_set',with_path=True)

data['label'] = data['path'].apply(lambda path: path.split('/')[-2])

model = tc.image_classifier.create(data, target='label', max_iterations=50)

model.export_coreml('Pets.mlmodel')
```

Conclusion

Machine learning is a very interesting but a challenging topic for many. High level libraries like Turi Create help democratize machine learning to the masses. However, I encourage anyone trying to get into machine learning to not just settle for high level libraries like Turi Create. While I believe that it is a great way to start your machine learning journey, Turi Create does not scale well when it comes to developing models with more than 1000 labels. Turi Create is a great solution for developing small applications and that is where the road ends.