



# 深度 | 详解苹果Core ML：如何为iOS创建机器学习应用？

2017年06月07日 12:15:26 机器之心

0

选自developer.apple

机器之心编译

参与：吴攀

在昨天开幕的 WWDC 2017 开发者大会上，苹果宣布了一系列新的面向开发者的机器学习 API，包括面部识别的视觉 API、自然语言处理 API，这些 API 集成了苹果所谓的 Core ML 框架；参阅机器之心报道《苹果开发者大会 WWDC 2017：首次全面展示苹果的人工智能实力》。软件主管兼高级副总裁 Craig Federighi 介绍说，Core ML 的核心是加速在 iPhone、iPad、Apple Watch 上的人工智能任务，支持深度神经网络、循环神经网络、卷积神经网络、支持向量机、树集成、线性模型等。大会之后，苹果开发者网站更新了相关文档，对 Core ML 进行了更加详细的介绍。机器之心在本文中对该文档的内容进行了翻译和整理。更多详情可访问原文档。

Core ML 文档地址：<https://developer.apple.com/documentation/coreml>

Core ML 介绍

使用 Core ML，你可以将训练好的机器学习模型整合到你的应用中。它支持以下操作系统：

iOS 11.0+Beta

macOS 10.13+Beta

tvOS 11.0+Beta

watchOS 4.0+Beta

训练好的模型（trained model）是将一个机器学习算法应用到一个训练数据集之后所得到的结果。然后该模型可以基于新的输入数据而进行预测。比如，如果一个模型在一个地区的房价数据上进行了训练，那么它就可能能够根据房子的卧室和浴室数量来预测房价。



Core ML 是域特定的框架和功能的基础。Core ML 支持用于图像分析的 Vision；用于自然语言处理的 Foundation（比如 NSLinguisticTagger 类）和用于评估已经学习到的决



机器之心

专业的人工智能媒体与产业服务平台。

## 热文排行

日榜

周榜

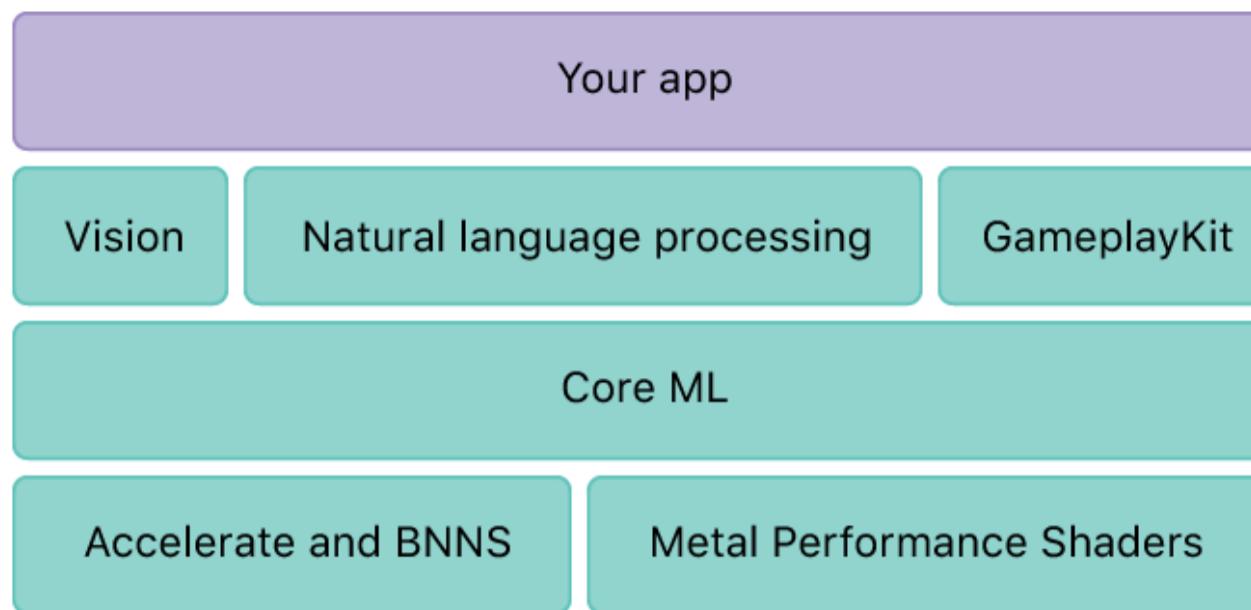
月榜

- 养老保险交15年和25年，领钱的时候差...
- 银行拼了！理财产品收益飙到7%，背后...
- 中国这一招灭了美国威风？原油定价权...
- 安迪做空包氏集团，金融圈大牙都要笑...
- 贾跃亭的中国好同学，今天公司一分钟...
- 洗牌下的中国城市战争将爆发，你在的...
- 最神秘高考教材老板：疑似牛散身家过...
- 一块饼干60元，杜海涛代言的减脂产品...
- 楼市的坏消息：房贷开始议价了！
- 韩国服软了！暂停部署萨德！



西安地铁问题电缆:处理结果公布 更多...

meplayKit。Core ML 本身构建于低层面的原语 ( primitives ) 之上 , 比如 Accelerate、BNNS 和 Metal Performance Shaders。



Core ML 为设备性能进行了优化 , 从而减少了内存占用和功耗。严格在设备上运行能够确保用户数据的隐私 , 并且能保证你的应用在没有网络连接时也能够工作和响应。

### 如何使用 Core ML ?

#### 获取 Core ML 模型

如何将 Core ML 模型用在你的应用中 ? Core ML 支持大量机器学习模型 , 包括神经网络、数集成、支持向量机和广义的线性模型。Core ML 需要 Core ML 格式 ( .mlmodel ) 。苹果已经以该格式提供了一些流行的开源模型 , 请访问 : <https://developer.apple.com/machine-learning> 。你可以下载这些模型 , 并用在你的应用中。另外 , 一些研究组和大学也会发布自己的模型和训练数据 , 这些可能并不是 Core ML 模型格式的。要使用这些模型 , 请将其转换成 Core ML 格式。

#### 将 Core ML 模型用在你的应用中

下面给出了一个示例 , 将一个训练好的模型 MarsHabitatPricer.mlmodel 用在了一个简单应用中 , 以用来预测火星上的地价。

##### 1. 将模型添加到你的 Xcode 项目

通过将模型拖拽到项目导航即可将该模型添加到你的 Xcode 项目。

Name	Type	Description
solarPanels	Double	Number of solar panels
greenhouses	Double	Number of greenhouses
size	Double	Size in acres
price	Double	Price of the habitat (in millions)

西安地铁问题电缆:处理结果公布 更多...



你可以在 Xcode 中打开模型来查看关于该模型的信息——包括模型的类型及其需要的输入和输出。该模型的输入是太阳能电池板和温室的数量，以及栖息地的大小（单位：英亩）。该模型的输出是该栖息地的预测价格。

## 2. 在代码中创建模型

Xcode 也会使用关于该模型输入和输出的信息来自动生成一个该模型的自定义的编程接口，你可以在代码中使用这些接口来和该模型进行交互。对于 MarsHabitatPricer.mlmodel，Xcode 会生成几个接口，其中 MarsHabitatPricer 表示该模型，MarsHabitatPricerInput 表示该模型的输入，MarsHabitatPricerOutput 则是该模型的输出。

使用生成的 MarsHabitatPricer 类的初始化器来创建该模型：

```
let model = MarsHabitatPricer()
```

## 3. 将输入值输入该模型

这个示例应用使用了 UIPickerView 来获取来自用户的输入值：

```
func selectedRow(for feature: Feature) - Int {  
  
    return pickerView.selectedRow(inComponent: feature.rawValue)  
  
}  
  
let solarPanels = pickerDataSource.value(for: selectedRow(for: .solarPanels),  
  
feature: .solarPanels)let greenhouses = pickerDataSource.value(for: selectedRow(for: .greenhouses),  
  
feature: .greenhouses)let size = pickerDataSource.value(for: selectedRow(for: .size), feature:  
  
feature: .size)
```

## 4. 使用模型来做预测

MarsHabitatPricer 类有一个生成的 prediction(solarPanels:greenhouses:size:) 方法 (method)，可用于根据模型的输入值预测一个价格。在这个案例中，输入是太阳能电池板和温室的数量，以及栖息地的大小（单位：英亩）。这个方法的结果是一个 MarsHabitatPricerOutput 实例 marsHabitatPricerOutput

```
guard let marsHabitatPricerOutput = try? model.prediction(solarPanels: solarPanels,  
  
greenhouses: greenhouses, size: size) else {  
  
    fatalError("Unexpected runtime error.")  
  
}
```

获取 marsHabitatPricerOutput 的 price 属性来获取预测的价格，并将结果展示在该应用的用户界面上：

```
let price = marsHabitatPricerOutput.price
```

```
priceLabel.text = priceFormatter.string(for: price)
```

注：生成的 prediction(solarPanels:greenhouses:size:) 方法可能会报错。你会遇到的最常见的错误类型是你输入该方法的数据的类型与该模型所需的输入数据类型不符——比如，格式不对的图片。在这个示例应用中，输入是 Double 类型的。任何类型不匹配都会在编译时显现，对于这个示例应用而言，如果有什么地方不对，它就会报出一个致命错误。

## 5. 构建和运行一个 Core ML 应用

Xcode 可以将该 Core ML 模型编译成一个资源（resource），其已经为设备上的运行进行了优化。这种优化了的模型的表示被包含在你的应用捆绑（app bundle）中，而且该应用在设备上运行时会使用这个模型来进行预测。

### 将训练好的模型转换成 Core ML

如果你的模型是使用支持的第三方机器学习工具创建和训练的，那么你可以使用 Core ML Tools（<https://developer.apple.com/machine-learning>）将其转换成 Core ML 模型格式。下表给出了支持的模型和第三方工具。

Core ML Tools 是一个 Python 包（coremltools），托管在 Python Package Index (PyPI) 上。要了解更多有关 Python 包的信息，请参阅：<https://packaging.python.org>

	模型类型	支持的模型	支持的工具
1	神经网络	前馈、卷积、循环	Caffe Keras 1.2.2
2	树集成	随机森林、boosted trees、决策树	scikit-learn 0.18 XGBoost 0.6
3	支持向量机	标量回归、多类分类	scikit-learn 0.18 LIBSVM 3.22
4	广义的线性模型	线性回归、logistic 回归	scikit-learn 0.18
5	特征工程	稀疏矢量化、密集矢量化、分类处理	scikit-learn 0.18
6	Pipeline 模型	顺序链式模型	scikit-learn 0.18

### 1. 转换你的模型

使用对应你的模型的第三方工具的 Core ML 转换器转换你的模型。调用该转换器的 convert 方法，并将结果保存为 Core ML 模型格式（.mlmodel）。比如，如果你的模型是用 Caffe 创建的，那么将该 Caffe 模型（.caffemodel）转递给 coremltools.converters.caffe.convert：

```
import coremltools
```

```
coreml_model = coremltools.converters.caffe.convert('my_caffe_model.caffemodel')
```

然后将所得到的模型保存为 Core ML 模型格式：

```
coreml_model.save('my_model.mlmodel')
```

根据你模型的需要，你可能需要更新输入、输出和标签，或者你可能需要声明图像名、类型、格式。这些转换工具捆绑了更多文档，因工具不同，其提供的选项也有所不同。

### 2. 你也可以编写自定义的转换工具

当你需要转换一个不在上表中的格式的模型时，你也可以创建你自己的转换工具。

编写你自己的转换工具涉及到将你的模型的输入、输出和架构的表示（represer

西安地铁问题电缆:处理结果公布 更多...

翻译成 Core ML 模型格式。你需要定义该模型架构的每一层以及它们与其它层的连接。使用 Core ML Tools 提供的转换工具为例；它们演示了通过第三方工具创建的多种类型的模型被转换成 Core ML 模型格式的方法。

注：Core ML 模型格式是由一些协议缓冲文件（protocol buffer files）定义的，具体描述请参阅：<https://developer.apple.com/machine-learning>

#### Core ML API

这个文档包含了正在开发中的 API 和相关技术的初步信息。这些信息可能会发生变化，遵循这个文档实现的软件应该在最后的操作系统软件上进行测试。了解更多使用苹果 Beta 软件的信息，访问：<https://developer.apple.com/support/beta-software/>

在大多数案例中，你仅需要与你的模型的动态生成的接口进行交互，这些接口是 Xcode 自动生成的。你可以使用 Core ML API 直接支持自定义的工作流和高级的用例。比如，如果你需要做预测，同时异步地将输入数据收集到一个自定义的结构中，你可以通过采用 MLFeatureProvider 协议来使用该结构来为你的模型提供输入特征。



<b>Model</b>	<b>class MLModel</b> An encapsulation of all the details of your machine learning model. <input type="button" value="Beta"/>	<b>+</b>
<b>Model Features</b>	<b>protocol MLFeatureProvider</b> An interface that represents a collection of feature values for a model. <input type="button" value="Beta"/>	<b>+</b>
	<b>class MLDictionaryFeatureProvider</b> A convenience wrapper for the given dictionary of data. <input type="button" value="Beta"/>	<b>+</b>
	<b>class MLFeatureValue</b> An immutable instance representing a feature's type and value. <input type="button" value="Beta"/>	<b>+</b>
	<b>class MLFeatureDescription</b> A description of a feature. <input type="button" value="Beta"/>	<b>+</b>
	<b>class MLMultiArray</b> A multidimensional array used as input or output for a model. <input type="button" value="Beta"/>	<b>+</b>
<b>Errors</b>	<b>struct MLModelError</b> Error codes for Core ML. <input type="button" value="Beta"/>	<b>+</b>

更多有关GMIS 2017大会的内容，请点击「[阅读原文](#)」查看机器之心官网↓↓↓



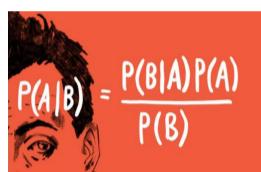
西安地铁问题电缆:处理结果公布 更多...



0

## 作者历史文章

### 从贝叶斯角度，看深度学习的属性和改进方法



选自arXiv.org机器之心编译参与：蒋思源、吴攀深度学习是一种高效的非线性高维数据处理方法，它可以更自然地解释为一种工程或算法，而本论文希望从贝叶斯的角度将[详细]

2017年 06月08日 12:15

### 关系推理水平超越人类：DeepMind展示全新神经网络推理预测技术



选自DeepMind作者：Adam Santoro等机器之心编译参与：机器之心编辑部想象一下在阿加莎·克里斯蒂（《东方快车谋杀案》作者）的侦探小说里收集所有证据找[详细]

2017年 06月07日 12:15

### 学界 | AAAI-17论文提出深度学习C语言修改器DeepFix：用人工智能加



选自aaai.org机器之心编译参与：吴攀如果你是一个程序员，你一定体验过编译时报错的痛苦，有时候一个小错误可能就需要你很多时间检查许多代码才能找到。近日，印度[详细]

2017年 06月07日 12:15

### 报名 | 图普科技「让世界充满 AI！」，与您相约浙大



这周6月8日，图普科技与您相约美丽的浙江大学2014年末，诺兰导演的一部《星际穿越》让无数影迷叹为观止，而全程人气最高的角色——智能冰箱更是让人耳目一新。“等下[详细]

2017年 06月07日 12:15

### 开源 | 基于Metal的机器学习框架Bender：可在iOS上运行TensorFlow模



选自GitHub机器之心编译参与：吴攀在正在举行的WWDC2017上，苹果宣布发布了Metal2，详情可参阅机器之心的报道《苹果开发者大会WWDC2017：首次[详细]

2017年 06月07日 12:15

### 专栏 | 大漠孤烟，长河落日：面向景深结构的风景照生成技术



机器之心专栏上海交通大学电子工程系作者：杨蕊简介2014年以来，生成对抗网络（Generative Adversarial Networks）已经在各式各样的图像生[详细]

[西安地铁问题电缆:处理结果公布](#) [更多...](#)

2017年 06月06日 06:15

## 苹果开发者大会WWDC 2017：首次全面展示苹果的人工智能实力



机器之心报道参与：李亚州、XavierMassa当地时间6月5日，苹果开发者年度盛会WWDC2017在美国加州举行。在这个舞台上，我们看到了苹果软件、硬件有哪些[详细]

2017年 06月06日 06:15

## 实验研究工作流程详解：如何把你的机器学习想法变成现实



选自dustintran作者：DUSTINTRAN 机器之心编译参与：李泽南、Smith从研究思想的提出到实验的具体实现是工程中的基础环节。但是这一过程常常被一[详细]

2017年 06月05日 13:15

## 开源 | 浏览器上最快的DNN执行框架WebDNN：从基本特性到性能测评



选自Github机器之心编译参与：蒋思源、晏奇WebDNN是网页浏览器中最快的DNN执行框架，而本文首先简单介绍了WebDNN特征与其框架结构，即表明了为什么W[详细]

2017年 06月05日 13:15

## 学界 | CMU新研究试图统一深度生成模型：搭建GAN和VAE之间的桥梁



选自arXiv机器之心编译参与：吴攀不同的深度生成模型之间存在怎样的共性？近日，来自CMU和Petuum的四位研究者ZhitingHu、ZichaoYang、R[详细]

2017年 06月05日 13:15