Core ML & Create ML

Machine Learning Framework

Overview

What is Machine Learning?

Core ML & Create ML

Live Coding Example

UI Use Cases & Discussion

Music Tagging

Face Detection

Siri

What is Machine Learning?

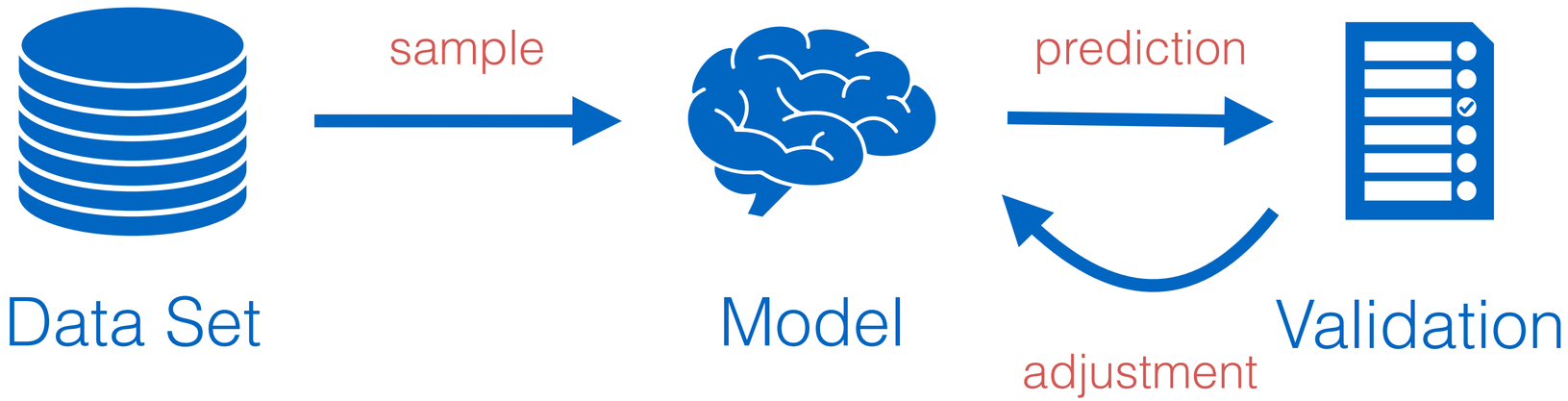
What is Machine Learning?

- Artificial Intelligence != Machine Learning
-

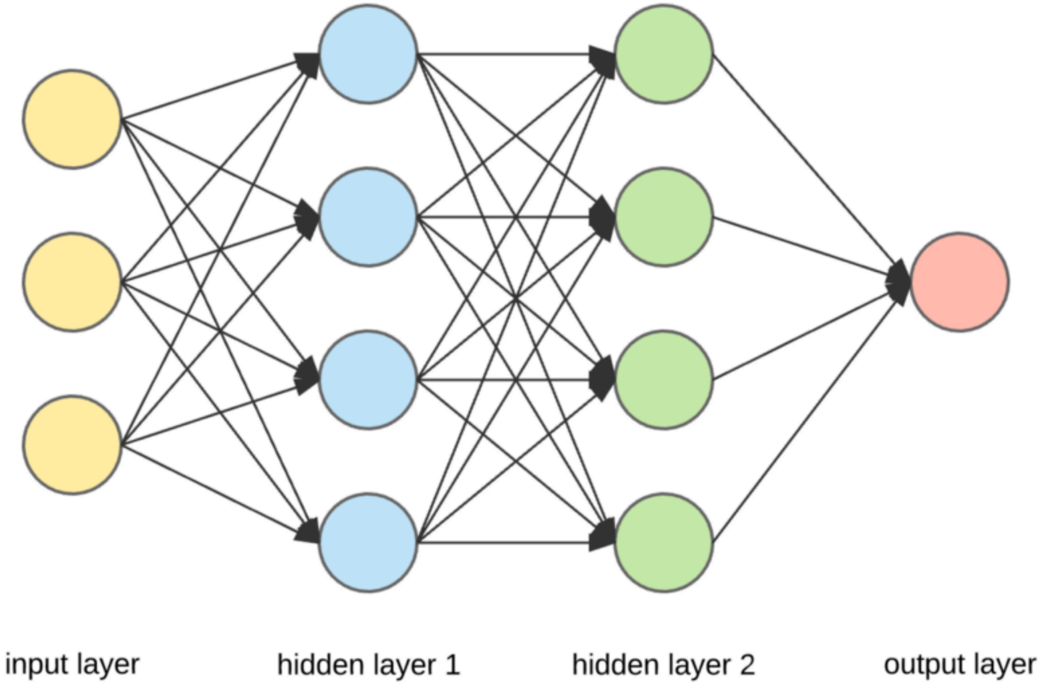
What is Machine Learning?

- Artificial Intelligence != Machine Learning
- Learning a function (from examples)

What is Machine Learning?



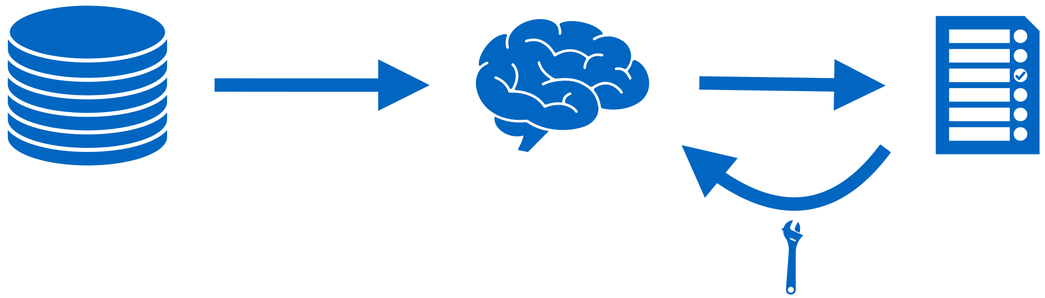
Neural Networks



https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

ML Methods

- **Supervised**
- Unsupervised
- Semi-supervised
- Reinforced
- ...

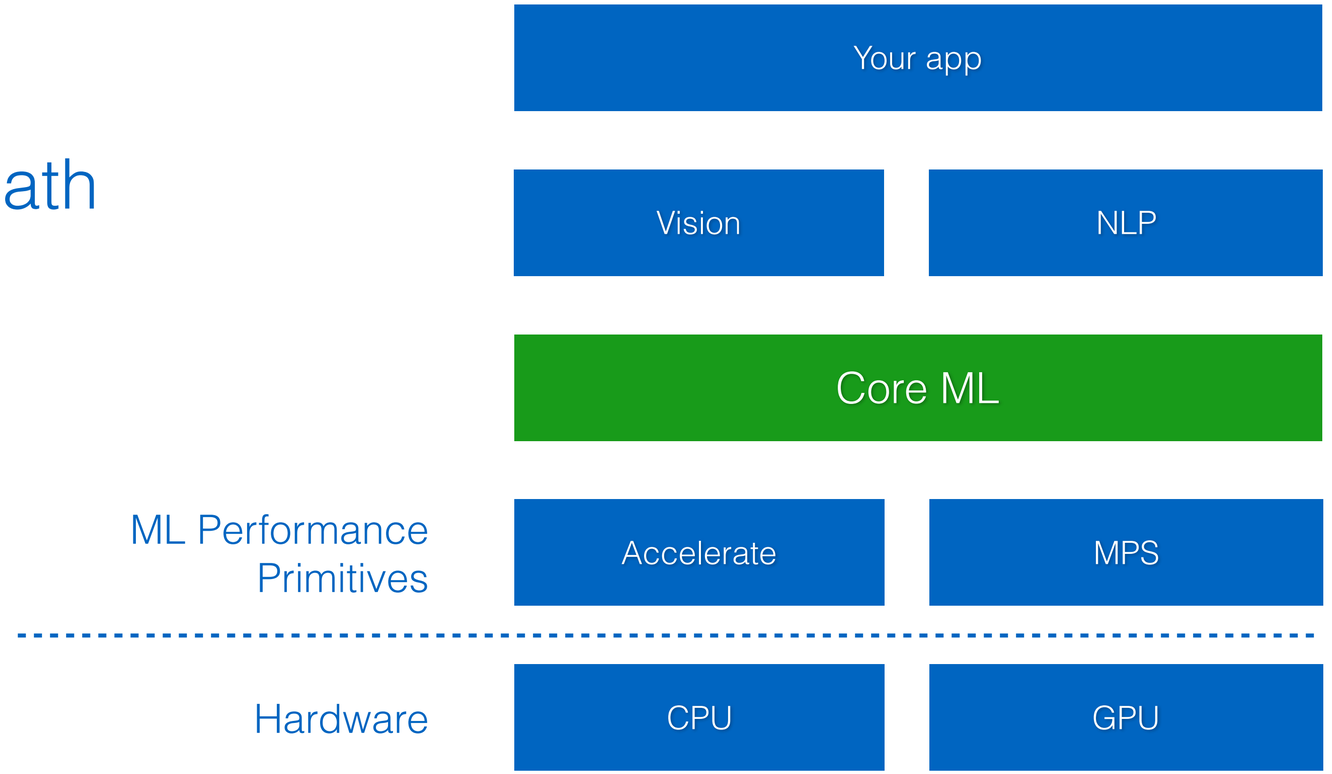


Machine Learning Framework

- **Core ML** (implement models)
- CoreMLTools (convert models)
- **Create ML** (create models)

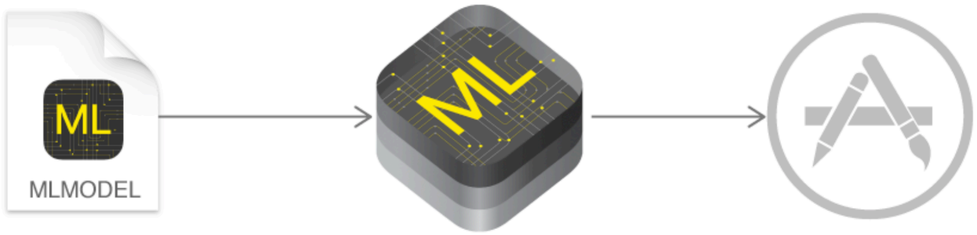
Architecture

- High performance math
- Context Switching
- CoreMLModel



Core ML

- Xcode integration
  - macOS, iOS, watchOS, tvOS
- Simple model integration & prediction



Basic Code

```
01 import CoreML
02
03
04
05
06
07
08
09
10
11
```

Basic Code

```
01 import CoreML
02
03 // Initialize model
04 let model = MyCoreMLModel()
05
06
07
08
09
10
11
```

Basic Code

```
01 import CoreML
02
03 // Initialize model
04 let model = MyCoreMLModel()
05
06 // Get user input & convert
07 ...
08
09
10
11
```

Basic Code

```
01 import CoreML
02
03 // Initialize model
04 let model = MyCoreMLModel()
05
06 // Get user input & convert
07 ...
08
09 // Make a prediction
10 let result = model.prediction(input: foo)
11 print(result.output)
```

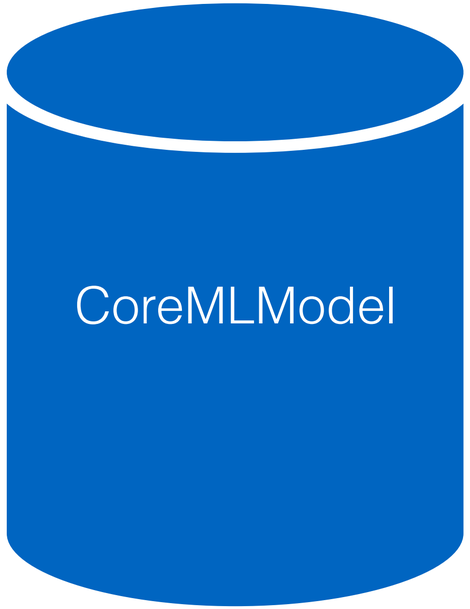
01

CoreML Model

- Datatypes
  - Double, Int, String
  - CVPixelBuffer, MLMultiArray, Dictionaries [String : Double]

- Classification
- Regression

CoreMLTools

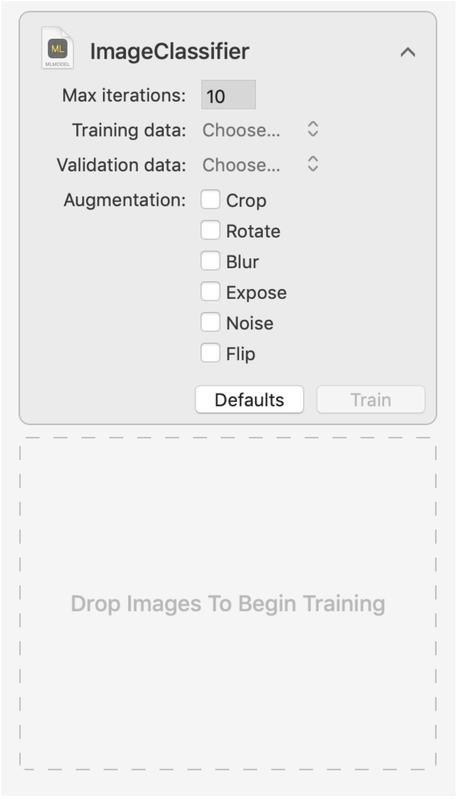


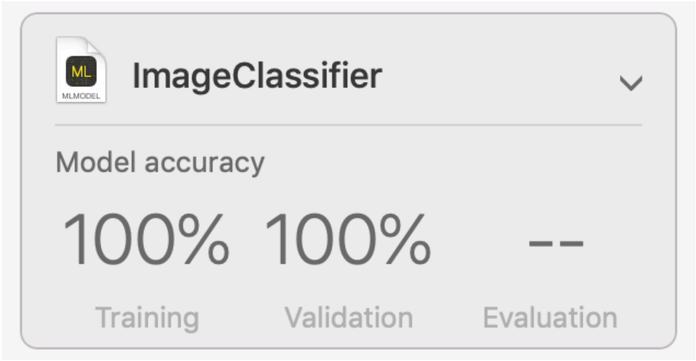
New in CoreML2

- CreateML
- Weight Quantization (reduce model size)
- Download and compile models
- Custom Model Layers (conversion)

Create ML

- Train Models
  - Image Classifier (Live UI Builder)
  - Text Classifier
  - Tabular Data
- Adjust training parameters
- Evaluation

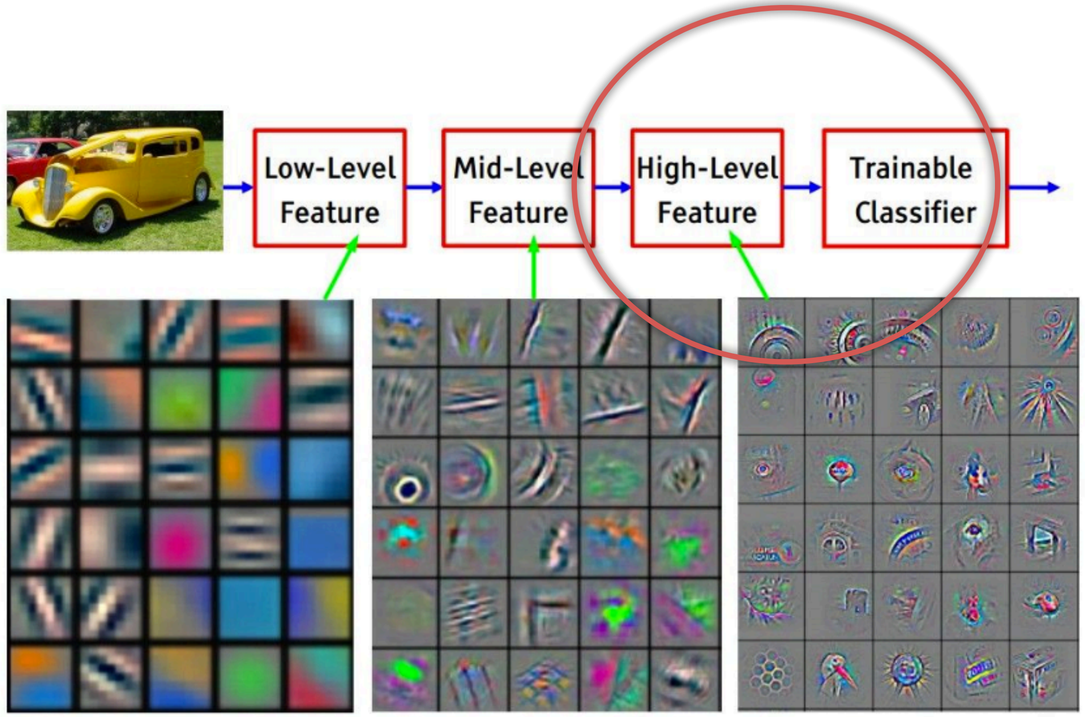


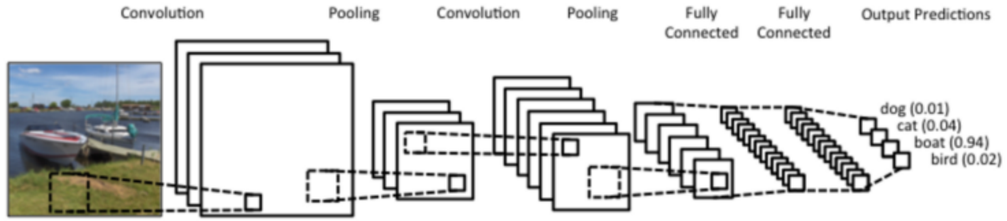


b1	b2	b3	b4	b5	b6	b7	b8	dec
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
0	0	0	0	0	1	0	1	5

Live Coding Demo

Transfer Learning





https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f

UI Use Cases
&
Discussion

UI Use Cases

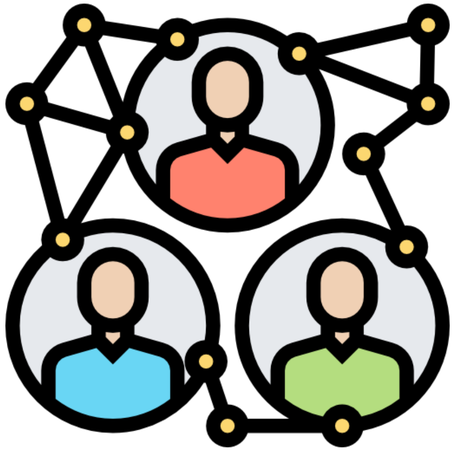
Music Tagging

Text Prediction

Siri

User Context & Model

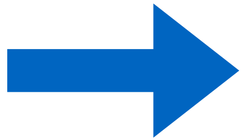
- User Agents
- Context Recognition
- Contextual Environments
- File System



User Context & Mental Model

from: anne@some-domain.de
subject: Project A

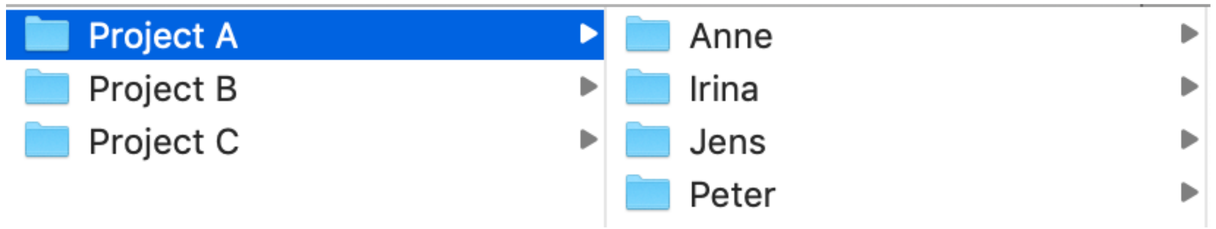
attachment: "pA_confidential.file"



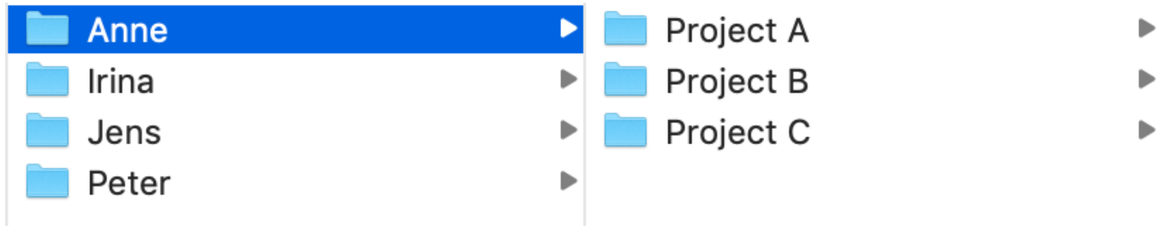
?

User Context & Mental Model

Project vs. Person

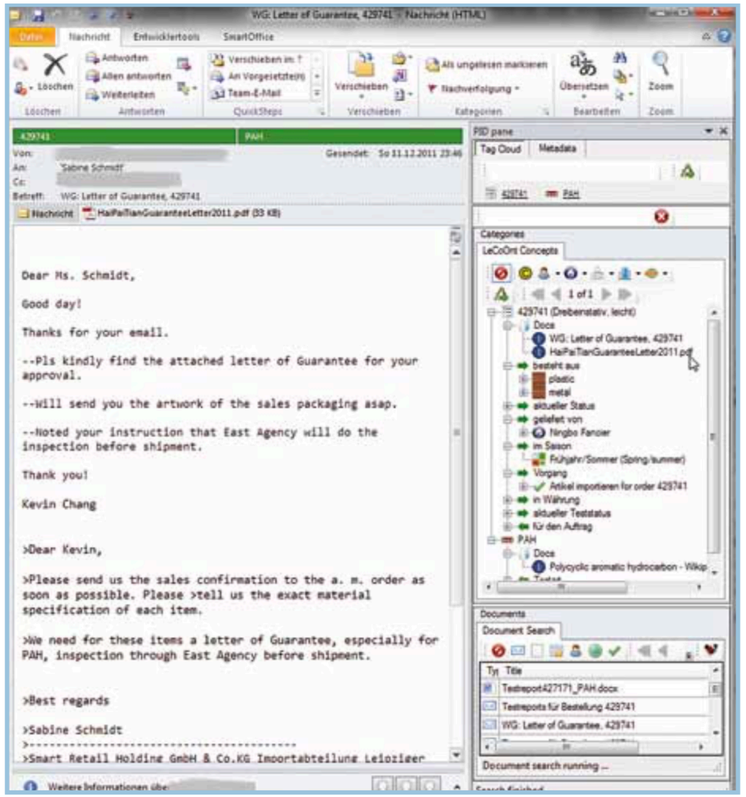


or



Semantic Desktop (DFKI)

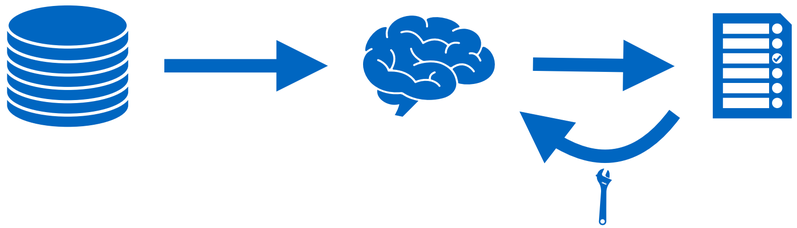
- Ontology
- How to present predictions?
- Adjust UI to Context?
- Intentional Forgetting



- Nepomuk (KDE-Project, Ontology)
- https://www.semiodesk.com/products/organiser/

Summary

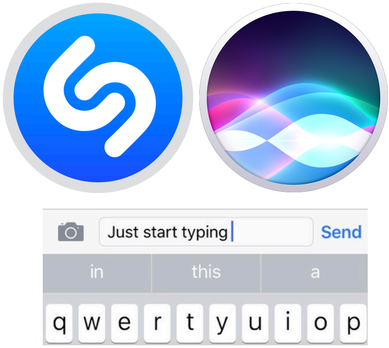
learning from examples



CreateML
CoreMLTools
CoreML

create
convert
implement)

users mental model



Sources

- http://developer.apple.com/machine-learning (good starting point for CoreML)
- https://developer.apple.com/videos/frameworks/machine-learning-and-vision (WWDC sessions)
- https://github.com/likedan/Awesome-CoreML-Models (great model database)
- http://alexsosn.github.io/ml/2017/06/09/Core-ML-will-not-Work-for-Your-App.html (disadvantages)
- http://alexsosn.github.io/ml/2015/11/05/iOS-ML.html (alternatives to CoreML)
- https://github.com/apple/turicreate (starting point for creating more advanced CoreML models)

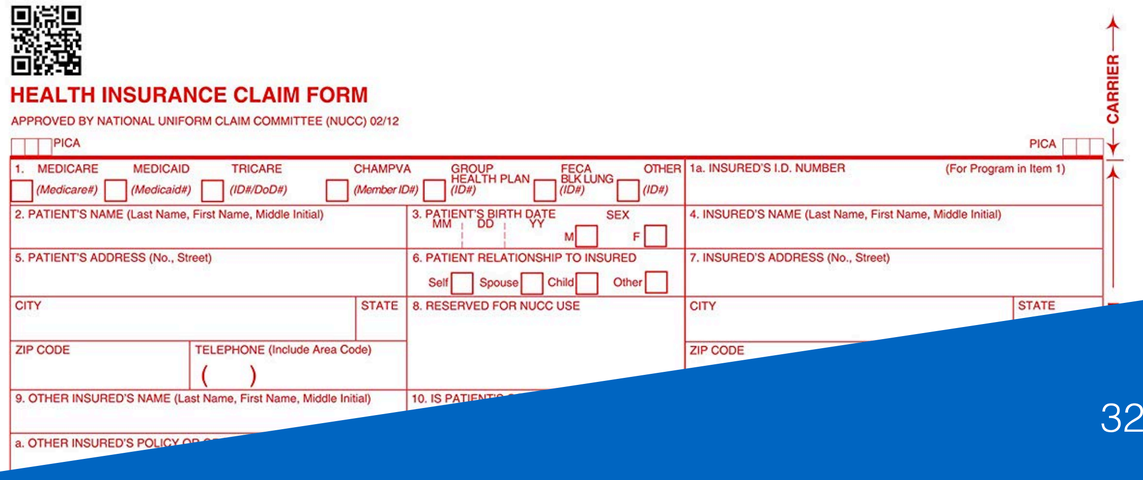
Sources

- https://pypi.org/project/coremltools/ (CoreML Tools Python package)
- https://www.raywenderlich.com/188-beginning-machine-learning-with-keras-core-ml (CoreML + Keras beginners guide)

- https://www.appcoda.com/coreml2/ (Whats new in CoreML2)
- https://talhassner.github.io/home/publication/2015_CVPR (age classifier)
- https://www.opendfki.de/trac/ (dfki open source projects)
- https://www.dfki.de/web/forschung/projekte-publikationen/projekte/projekt/nepomuk/ (NEPOMUK project)

Sources

- https://aws.amazon.com/marketplace/search/results?page=1&filters=fulfillment_options&fulfillment_options=SAGEMAKER (Amazon ML models marketplace, interesting..)
- https://www.amazon.com/Claim-ICD-10-Version-02-12/dp/B0151ZFNEQ (insurancce claim form)


HEALTH INSURANCE CLAIM FORM

APPROVED BY NATIONAL UNIFORM CLAIM COMMITTEE (NUCC) 02/12

PICA

1. MEDICARE (Medicare#) MEDICAID (Medicaid#) TRICARE (ID#/DoD#) CHAMPVA (Member ID#) GROUP HEALTH PLAN (ID#) FECA BLK LUNG (ID#) OTHER (ID#)

1a. INSURED'S I.D. NUMBER (For Program in Item 1)

2. PATIENT'S NAME (Last Name, First Name, Middle Initial)

3. PATIENT'S BIRTH DATE MM DD YY SEX M F

4. INSURED'S NAME (Last Name, First Name, Middle Initial)

5. PATIENT'S ADDRESS (No., Street)

6. PATIENT RELATIONSHIP TO INSURED Self Spouse Child Other

7. INSURED'S ADDRESS (No., Street)

CITY STATE

8. RESERVED FOR NUCC USE

CITY STATE

ZIP CODE TELEPHONE (Include Area Code) ( )

ZIP CODE

9. OTHER INSURED'S NAME (Last Name, First Name, Middle Initial)

10. IS PATIENT'S

a. OTHER INSURED'S POLICY OR

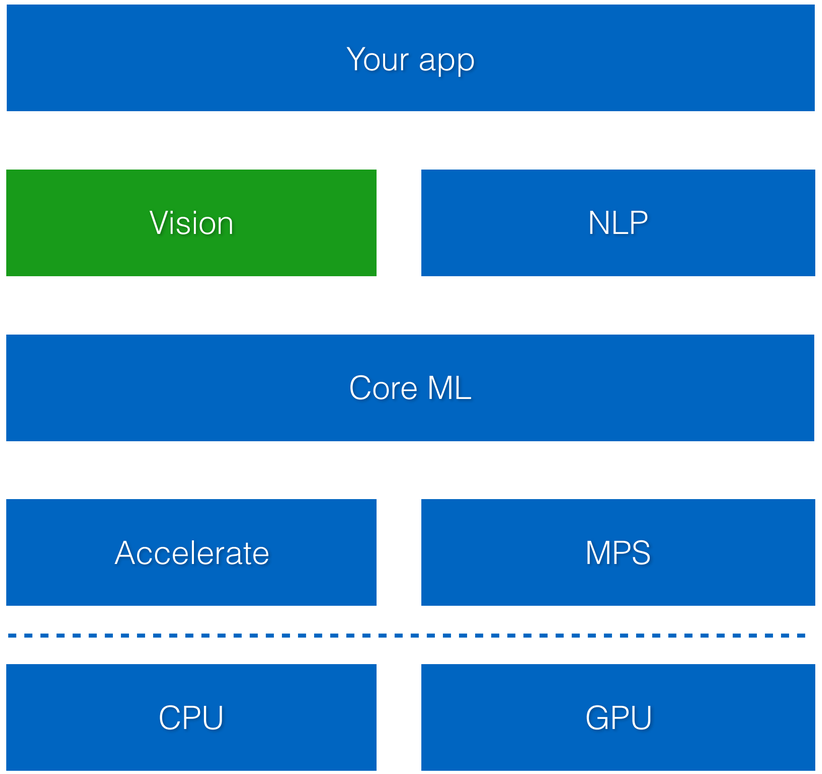
CARRIER

Bonus Backup

Vision Framework

- Image processing
  - Face detection, rectangle tracking ..
  - Optimized for accuracy

- Imaging pipeline
- request & completion handlers



Vision Code

```
01  import Vision
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
```

Vision Code

```
01  import Vision
02
03  // Initialize model
04  let model = VNCoreMLModel(for: MyCoreMLModel().model)
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
```

Vision Code

```
01  import Vision
02
03  // Initialize model
04  let model = VNCoreMLModel(for: MyCoreMLModel().model)
05
06
07  // Get interesting user input
08  ...
09
10
11
12
13
14
15
16
17
18
19
```

Vision Code

```
01  import Vision
02
03  // Initialize model
04  let model = VNCoreMLModel(for: MyCoreMLModel().model)
05
06
07  // Get interesting user input
08  ...
09
10  // Make a prediction
11  let request = VNCoreMLRequest(model: model, completionHandler: myResultsMethod)
12  let handler = VNImageRequestHandler(visionSupportedInput: foo)
13  handler.perform([request])
14
15
16
17
18
19
```

Vision Code

```
01  import Vision
02
03  // Initialize model
04  let model = VNCoreMLModel(for: MyCoreMLModel().model)
05
06
07  // Get interesting user input
08  ...
09
10  // Make a prediction
11  let request = VNCoreMLRequest(model: model, completionHandler: myResultsMethod)
12  let handler = VNImageRequestHandler(visionSupportedInput: foo)
13  handler.perform([request])
14
15  // Handle results
16  func myResultsMethod(request: VNRequest, error: Error?) {
17      let results = request.results as? [VNClassificationObservation]
18      print(results.first.output)
19  }
```