

移动设备也能玩机器学习

杨鑫奇 @xinqiyang
億百國際株式会社

Who am I?

杨鑫奇

wechat/weibo/twitter/facebook/github

- 全栈工程师 (PHP,Java,Object-C,Python,javascript,etc) @xinqiyang
- 曾任职于百度，干过派遣，现在自己折腾创业
- 现在东京创办亿咖啡，一个华人技术交流社区
- Founder of 億百國際株式会社

事先说明

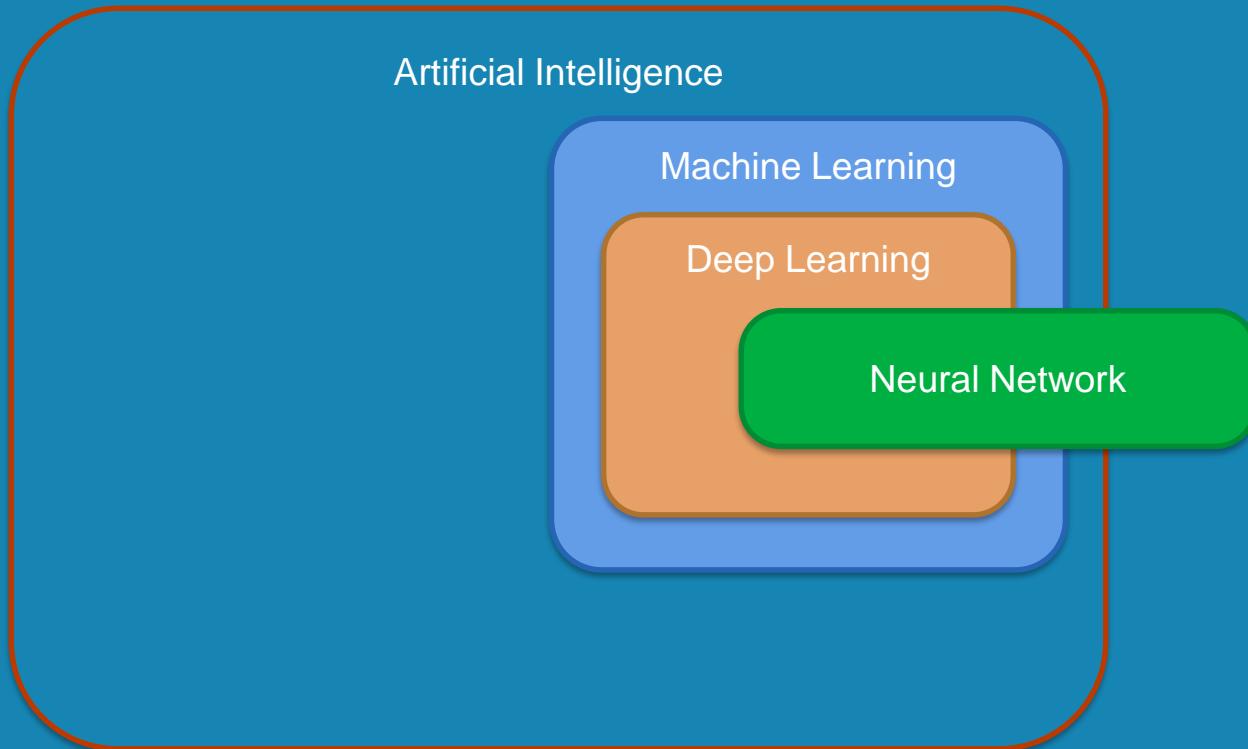
- 不是一个ML专家
- 不是资深Pythoner
- 也不太懂高深的算法
- 在ML领域涉世未深

那为什么还会来讲这一场？

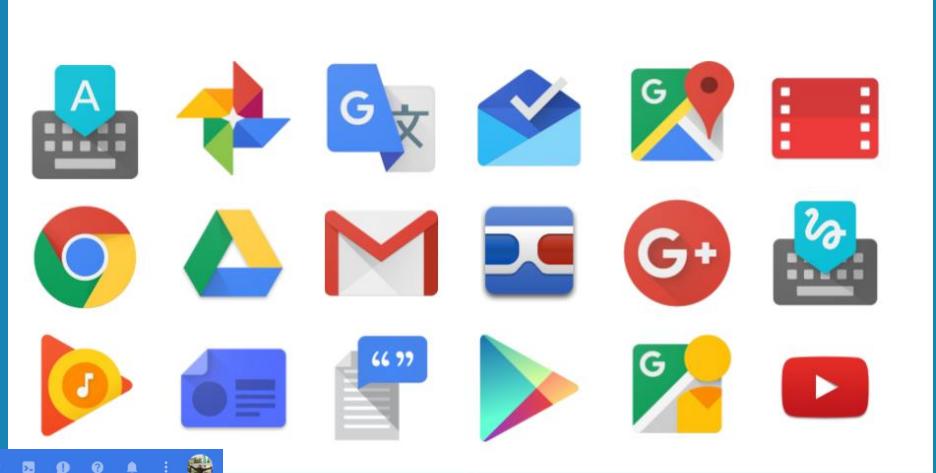
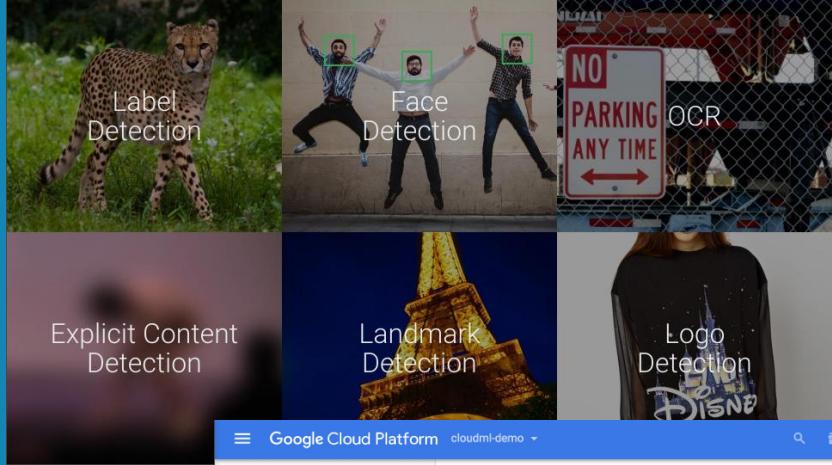
本次大纲

- 众观下各大厂商所提供的ML服务 (google,aws,microsoft, azure, aliyun,qcoud etc)
- 从移动方向看ML的应用场景具有极大的商业价值
- 从实际应用的角度来看Machine Learning
- 如何动手玩转ML
- 我们如何在APP中使用ML (演示)
- 如何用好ML工具来训练并使用自己的模型
- 移动设备上ML的训练和使用的未来会是什么样的？
- 提问回答 && 讨论

热门的热门，我们应该学什么？



各大厂商所提供的ML服务 - Google



Google Cloud Platform cloudmi-demo

ML Engine

Models

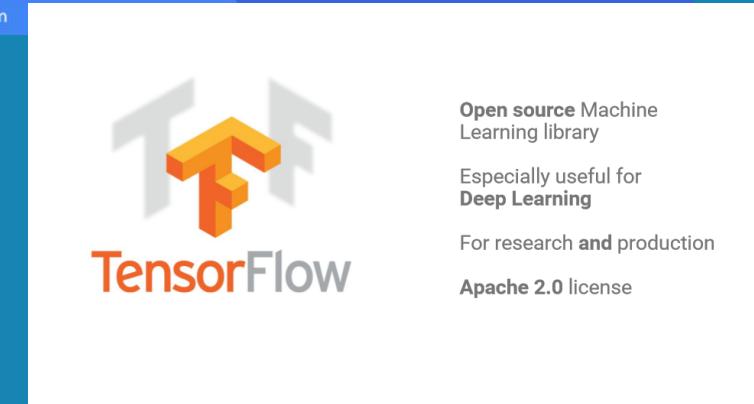
+ CREATE MODEL

Jobs

Models

Name	Default version
wnd1	vCMD2

Google Cloud



各大厂商所提供的ML服务 - AWS



AWS INNOVATE

Amazon Artificial Intelligence (AI)

Amazon
Rekognition

Amazon
Polly

Amazon
Lex

AI Services

Amazon
Machine Learning

Amazon
EMR

Spark &
Spark ML

AI Platforms

Apache
MXNet

TensorFlow

Caffe

Torch

Theano

CNTK

Keras

AI Engines

各大厂商所提供的ML服务 – Microsoft Azure

Security & Management

- Security Center
- Portal
- Azure Active Directory
- Azure AD B2C
- Multi-Factor Authentication
- Automation
- Scheduler
- Key Vault
- Store/ Marketplace
- VM Image Gallery & VM Depot

Platform Services

Media & CDN

- Media Services
- Media Analytics
- Content Delivery Network

Integration

- API Management
- BizTalk Services
- Logic Apps
- Service Bus

Compute Services

- Container Service
- VM Scale Sets
- Batch
- RemoteApp
- Dev/Test Lab

Application Platform

- Web Apps
- Mobile Apps
- API Apps
- Cloud Services
- Service Fabric
- Notification Hubs
- Functions

Data

- SQL Database
- SQL Server Stretch Database
- Redis Cache
- Storage Tables
- Azure Search
- DocumentDB

Intelligence

- Cognitive Services
- Bot Framework
- Cortana

Analytics & IoT

- HDIgnite
- Machine Learning
- Stream Analytics
- Data Catalog
- Data Lake Analytics Service
- Data Lake Store
- IoT Hub
- Event Hubs
- Data Factory
- Power BI Embedded

Hybrid Cloud

- Azure AD Health Monitoring
- AD Privileged Identity Management
- Domain Services
- Backup
- Operational Analytics
- Import/Export
- Azure Site Recovery
- StorSimple

Infrastructure Services

Compute

- Virtual Machines
- Containers

Storage

- Blob
- Queues
- Files
- Disks

Networking

- Virtual Network
- Load Balancer
- DNS
- Express Route
- Traffic Manager
- VPN Gateway
- App Gateway

Datacenter Infrastructure (34 Regions, 30 Online)

The bottom section shows a grid of server racks representing the datacenter infrastructure.

各大厂商所提供的ML服务 - AliYun

阿里云

全部导航 最新活动 产品 解决方案 数据·智能 安全 云市场 支持 合作伙伴 免费注册

GPU云服务器

GPU云服务器是基于GPU应用的计算服务，多适用于视频解码、图形渲染、深度学习、科学计算等应用场景，该产品具有实时高速，并行计算跟浮点计算能力等特点。

GPU实例用户调研问卷，欢迎大家填写[查看详情~](#)

GA1立即购买 GN4立即购买 GN5立即购买

图形

工程设计,非线性编辑
远程教育应用,3D展示

视频渲染

大规模高清视频转码,4K直播
多人视频会议,视频信号处理

计算

影视动画渲染,数字图像处理
计算金融,基因工程,科学计算

深度学习

图像处理识别,语音识别
视频内容鉴别,片源修复

深度学习 视频渲染 计算 图形

各大厂商所提供的ML服务 - QCloud

腾讯云 云产品 解决方案 云市场 合作与生态 文档与支持

English 免费体验 备案 登录 注册有礼 管理中心

计算

- 云服务器
- GPU 云服务器**
- FPGA 云服务器
- 专用宿主机
- 黑石物理服务器
- 黑石物理服务器
- 黑石ARM服务器
- 云硬盘
- 容器服务
- 弹性伸缩
- 负载均衡
- 私有网络
- 消息服务
- 消息服务CMQ
- Kafka
- 无服务器云函数

存储

数据库

数据处理

网络

GPU 云服务器

GPU 云服务器 (GPU Cloud Computing) 是基于GPU的应用于视频编解码、深度学习、科学计算等多种场景的快速、稳定、弹性的计算服务，我们提供和标准云服务器一致的管理方式。出色的图形处理能力和高性能计算能力为您提供极致计算性能，有效解放您的计算压力，提升产品的计算处理效率与竞争力。

[立即选购](#) [产品价格](#)

GPU 云服务器仅提供有限区域购买，请参考购买指引>>

产品优势

产品功能

应用场景

文档

产品优势

优势	GPU 云服务器	自建 GPU 物理服务器
弹性	<p>弹性扩展，灵活配置</p> <ul style="list-style-type: none">您只需几分钟时间即可轻松获取一个或若干个高性能计算实例；您可按需灵活定制，一键升级到更高性能和容量的实例规格，实现快速、平滑扩容，满足业务快速发展需要。	机器固定配置，难以满足变化的需求。
性能	<p>高效计算，极致性能</p> <ul style="list-style-type: none">透传GPU性能，极致发挥GPU性能；单机峰值计算能力突破 14T Flops 单精度浮点运算, 0.4T Flops 双精度浮点运算。	<ul style="list-style-type: none">用户手工容灾，依赖于硬件健壮性；数据物理单点，数据安全不可控。

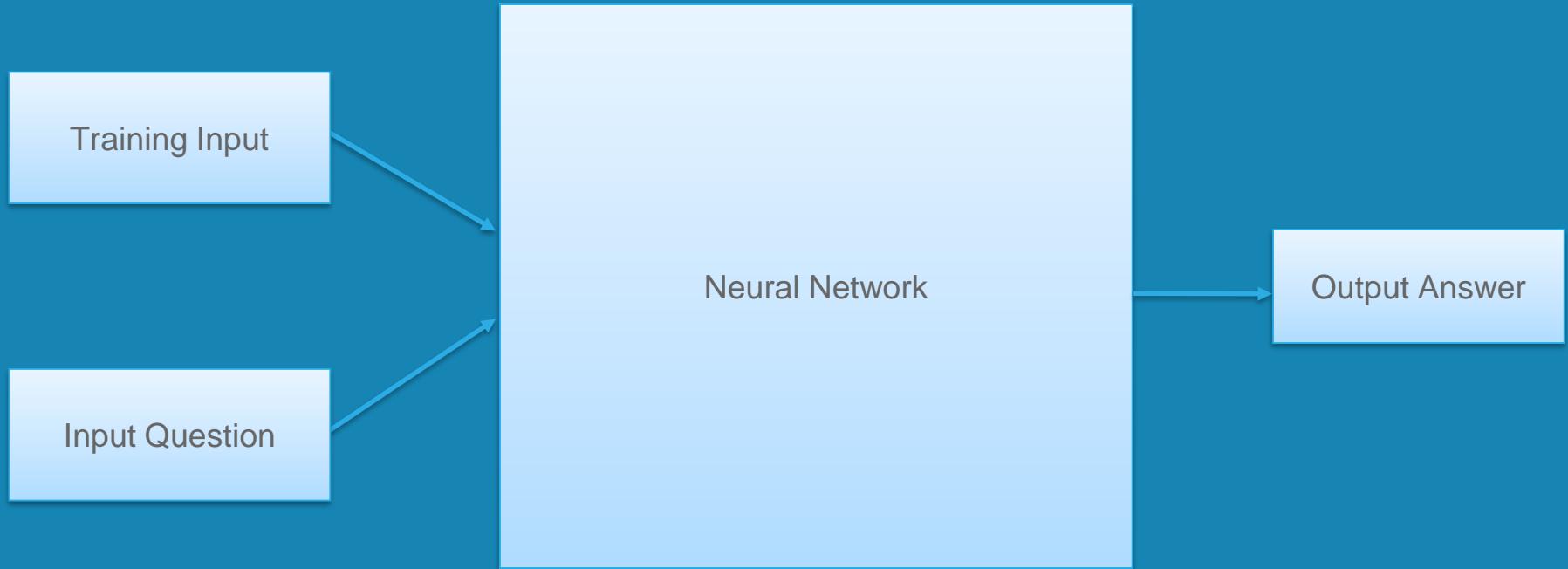
从移动方向看ML的应用场景具有极大的商业价值



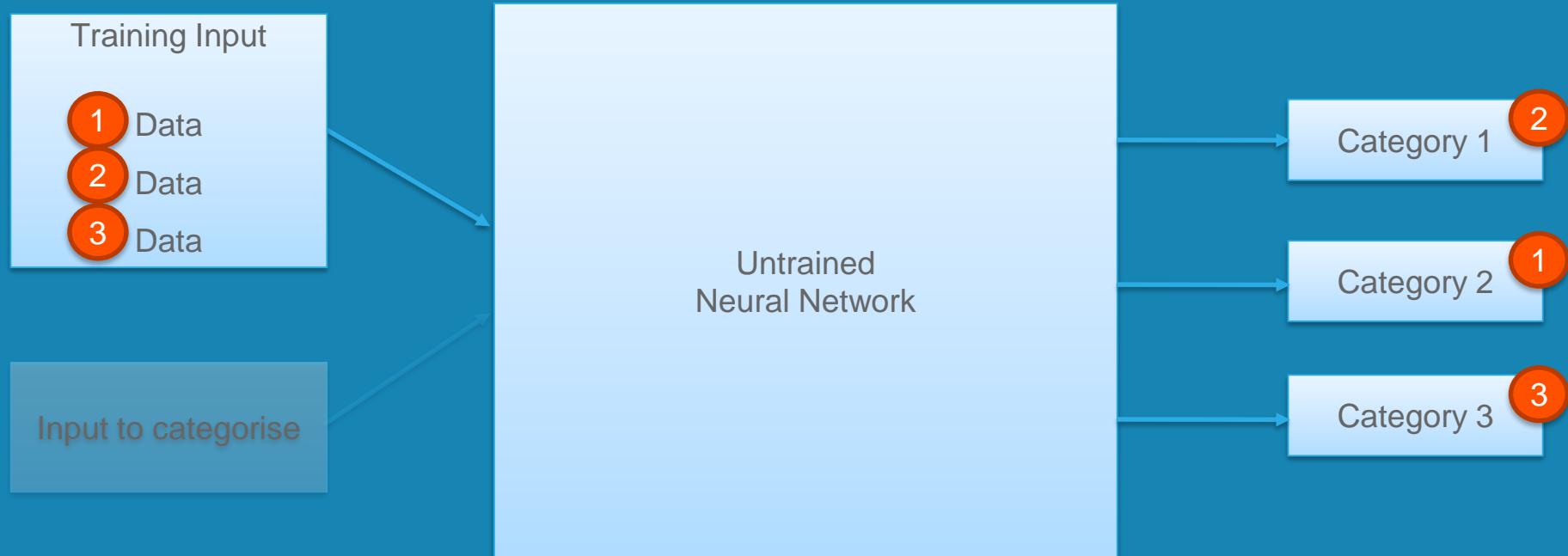
从实际应用的角度来看Machine Learning

- Spam
- Recommendations
- Handwriting recognition
- Speech recognition
- Face Detection
- Entity extraction
- Facial Recognition
- Object Recognition
- Text Prediction
- Sentiment Analysis
- Image Style transfer

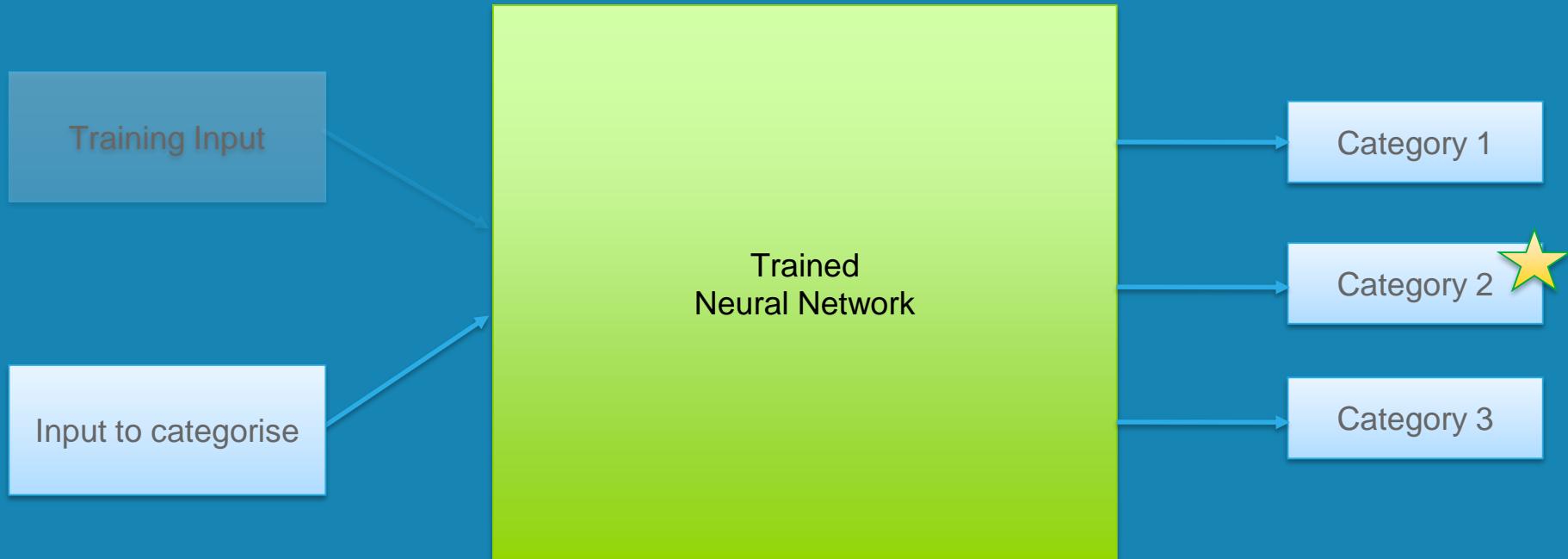
ML的目标



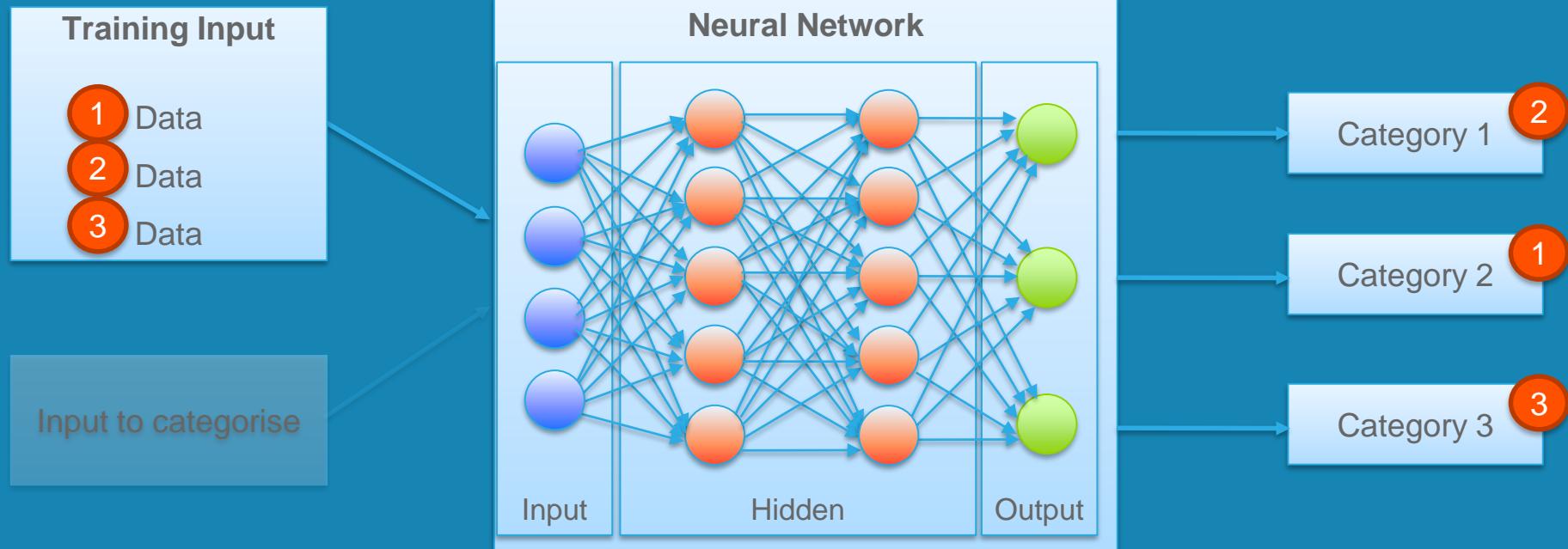
分类 Classifier



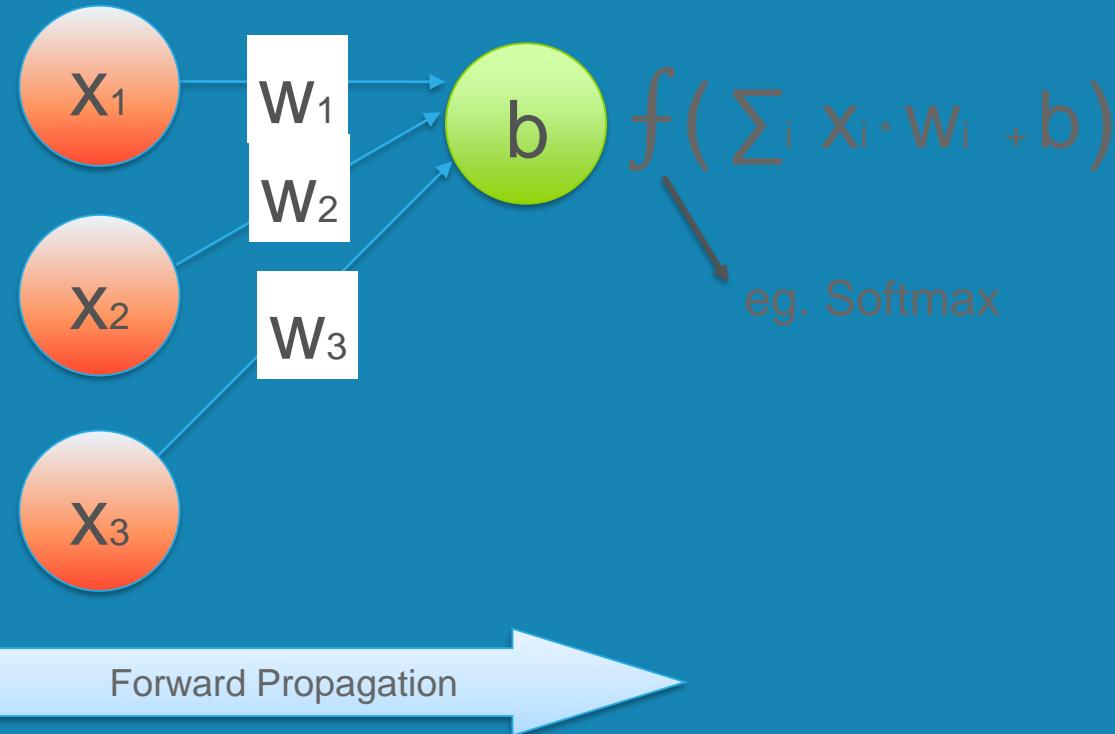
分类 Classifier



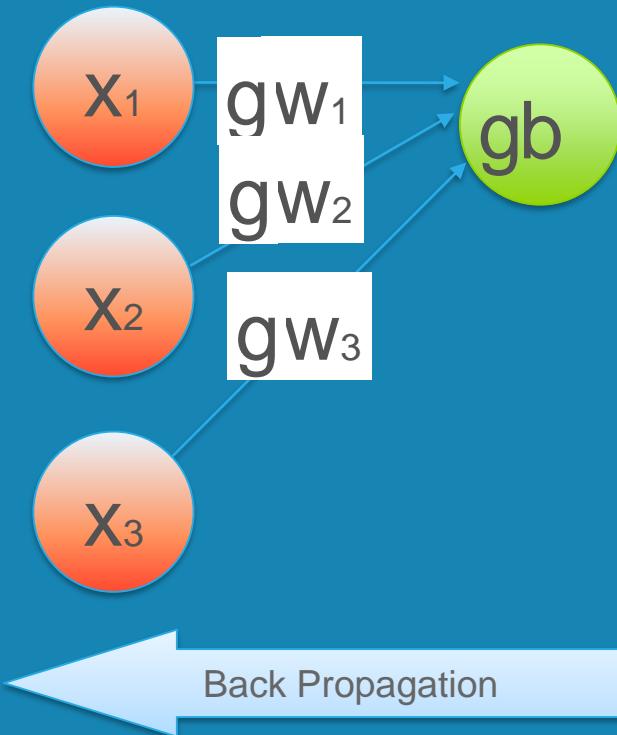
分类 Classifier



训练模型 Training the model



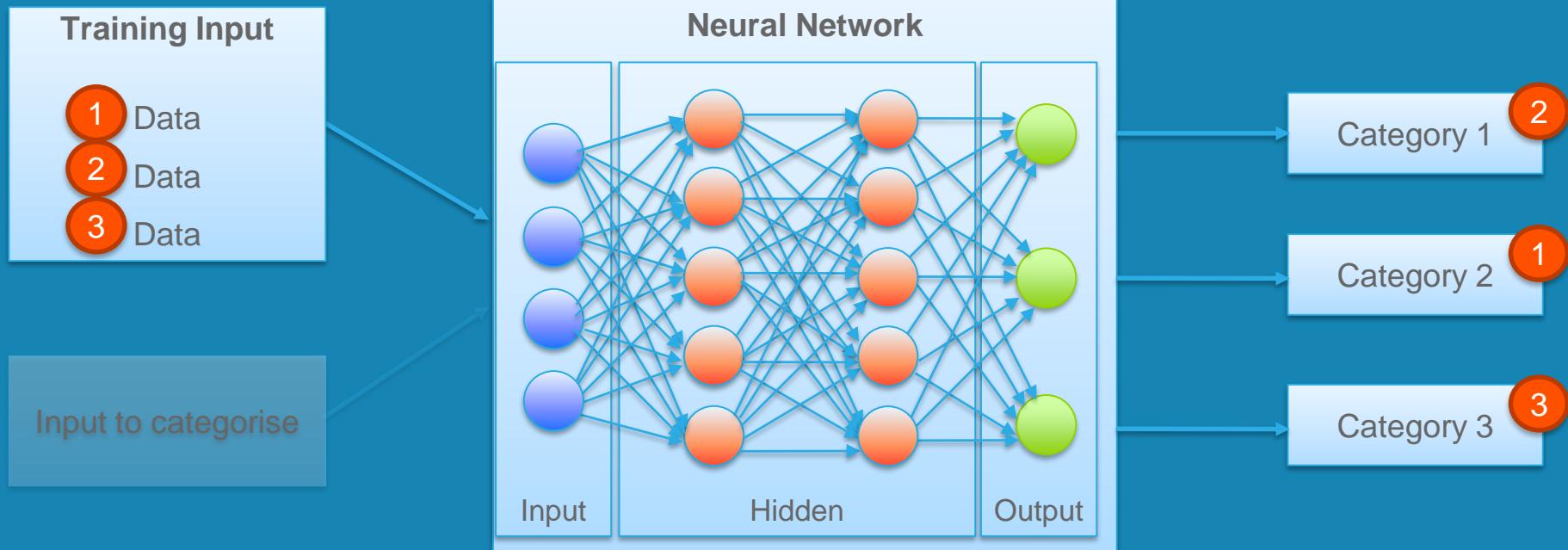
训练模型 Training the model



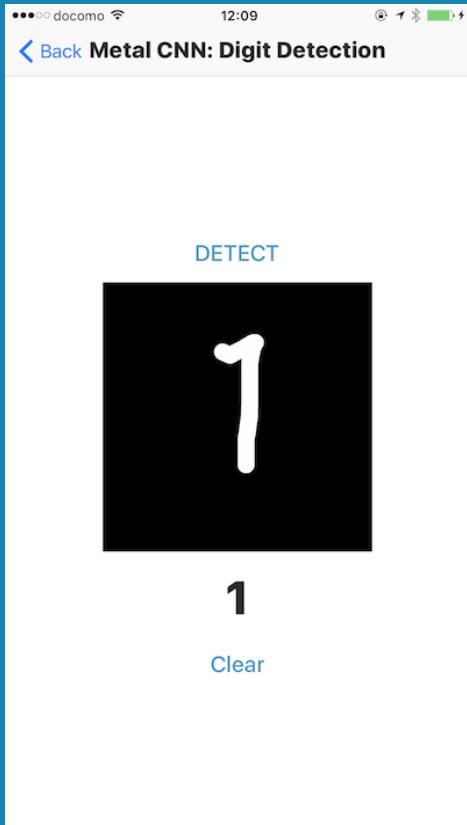
Difference between
expected and actual
= error

g = gradient
The extent to which
changing the value
reduces the error

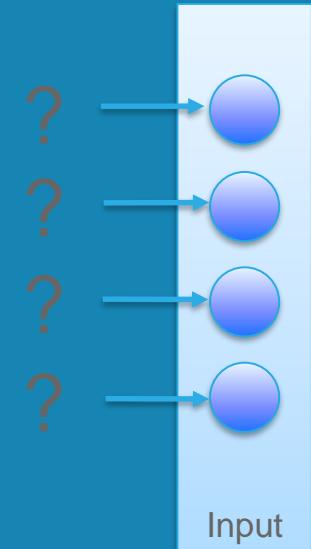
分类 Classifier



实例演示 手写图片识别 Handwriting Image Input

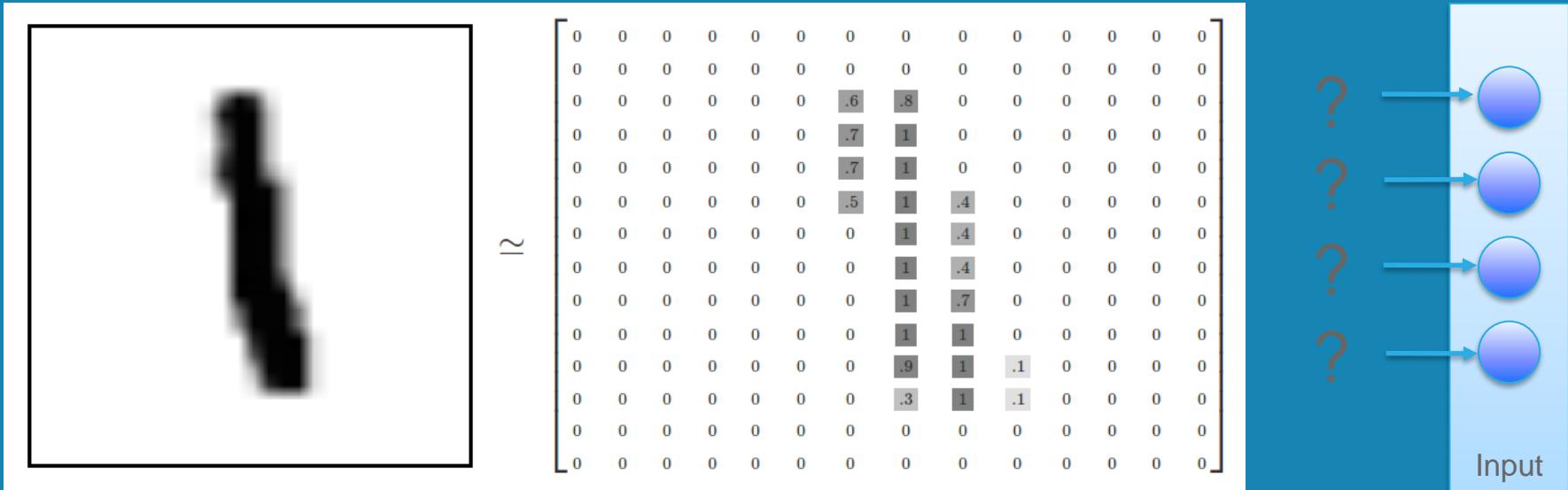


0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9



from the MNIST dataset

手写输入 Handwriting Image Input



14 pixels x 14 pixels

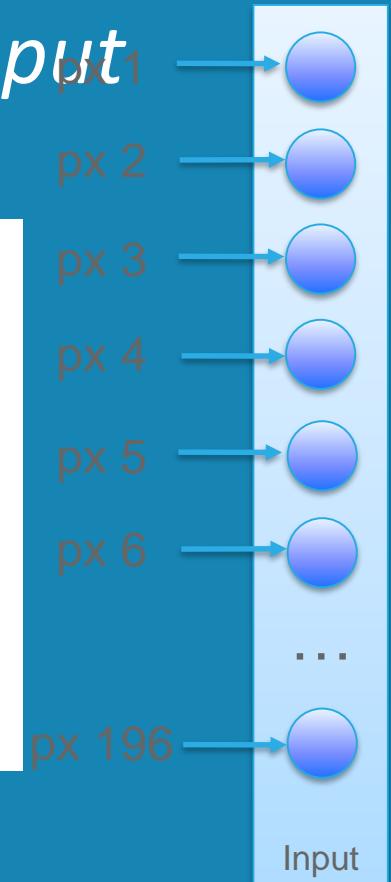
$14 \times 14 = 196$ values
between 0 and 1

手写图片输入 Handwriting Image Input



14 pixels x 14 pixels

$14 \times 14 = 196$ values
between 0 and 1



```
LaunchScreen.storyboard
Main.storyboard
AppDelegate.swift
RootViewController.swift
RootViewCell.swift
SampleDataSource.swift
Common
    UIViewController+alert.swift
Samples
    LivePhotoCapture
    UserNotification
    SpeechRecognition
    Looper
    PropertyAnimator
MetalCNN
    Common
    Basic
        DrawView.swift
        MNISTDeepCNN.swift
        MetalCNBasicViewController.swift
        MetalCNBasic.storyboard
ImageRecognition
    Inception3Net.swift
    Inception3Net+result.swift
    MetalImageRecognitionViewController.swift
    MetalImageRecognition.storyboard
AttributedSpeech
New Image Filters
Audio Fade-in/out
StickerPack
PersistentContainer
TabBadge
New Fonts
PreviewInteraction
Proactive
Haptic
Resources
    plist
        iOS10AddedFonts.plist
network_params
    batch_normalized_binaries
        bias_conv.dat
}

- Returns:
Guess of the network as to what the digit is as UInt
*/
func getLabel(finalLayer: MPSImage) -> UInt {
    // even though we have 10 labels outputted the MTLTexture format used is RGBAFloat16 thus 3 slices will
    // have 3*4 = 12 outputs
    var result_half_array = [UInt16](repeating: 6, count: 12)
    var result_float_array = [Float](repeating: 0.3, count: 10)
    for i in 0...2 {
        finalLayer.texture.getBytes(&result_half_array[4*i]),
            bytesPerRow: MemoryLayout<UInt16>.size*1*4,
            bytesPerImage: MemoryLayout<UInt16>.size*1*1*4,
            from: MTLRegion(origin: MTLOrigin(x: 0, y: 0, z: 0),
                            size: MTLSize(width: 1, height: 1, depth: 1)),
            mipmapLevel: 0,
            slice: i)
    }

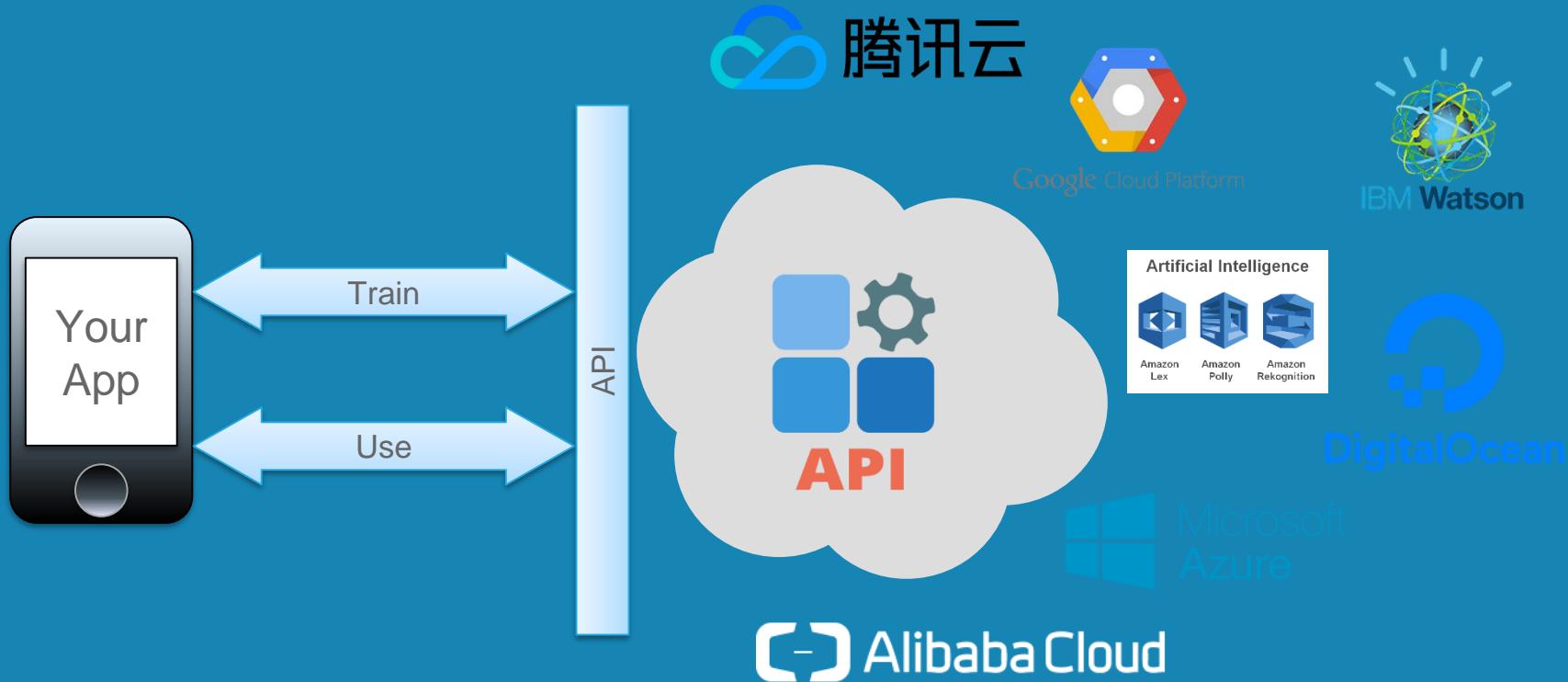
    // we use vImage to convert our data to float16, Metal GPUs use float16 and swift float is 32-bit
    var fullResultVImagebuf = vImage_Buffer(data: &result_float_array, height: 1, width: 10, rowBytes: 10*4)
    var halfResultVImagebuf = vImage_Buffer(data: &result_half_array, height: 1, width: 10, rowBytes: 10*2)

    if vImageConvert_Planar16FtoPlanarF(&halfResultVImagebuf, &fullResultVImagebuf, 0) != kvImageNoError {
        print("Error in vImage")
    }

    // poll all labels for probability and choose the one with max probability to return
    var max:Float = 0
    var mostProbableDigit = 10
    for i in 0...9 {
        if(max < result_float_array[i]){
            max = result_float_array[i]
            mostProbableDigit = i
        }
    }

    return UInt(mostProbableDigit)
}
```

如何动手玩转ML – 服务器端模式



如何动手玩转ML

调用云端提供的ML API

优点

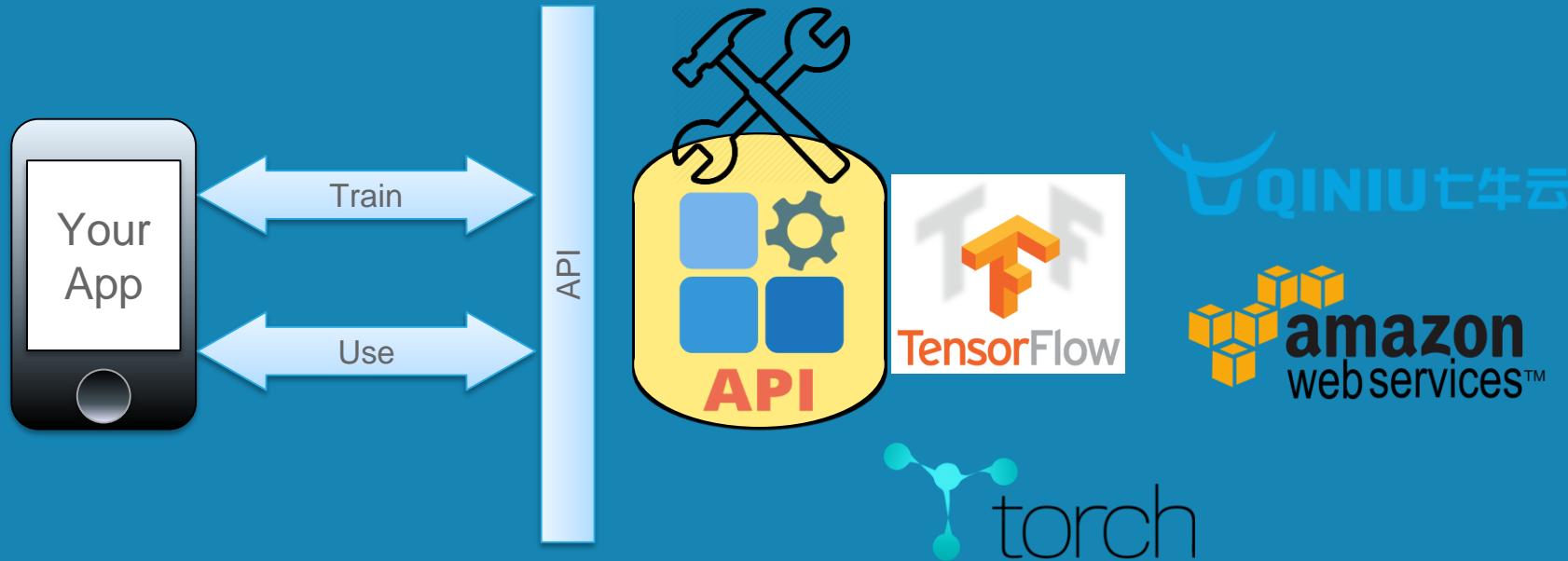
- 不需要ML知识
- 调用SDK简单实现
- 轻量灵活的资源占用

缺点

- 只能解决常见的ML问题
- 第三方资源依赖
- 需要一直在线
- 数据的所有权问题
- 无法做模型优化

如何动手玩转ML – 定制自己的后端

准备一台牛X机器开始自己训练之旅



如何动手玩转ML – 定制自己的后端

小杨，掏银子吧，出血了心疼了吧！

优点

- 可以自定义自己需要的模型
- 可以使用很多开源的模型
- 移动端无需占用资源
- 数据完全可控
- 可以做模型优化

缺点

- 需要掌握ML引擎及工具
- 还得会Python
- 需要自己管理服务器资源占用
- 自己构建API接口给客户端
- 需要私有服务器在线



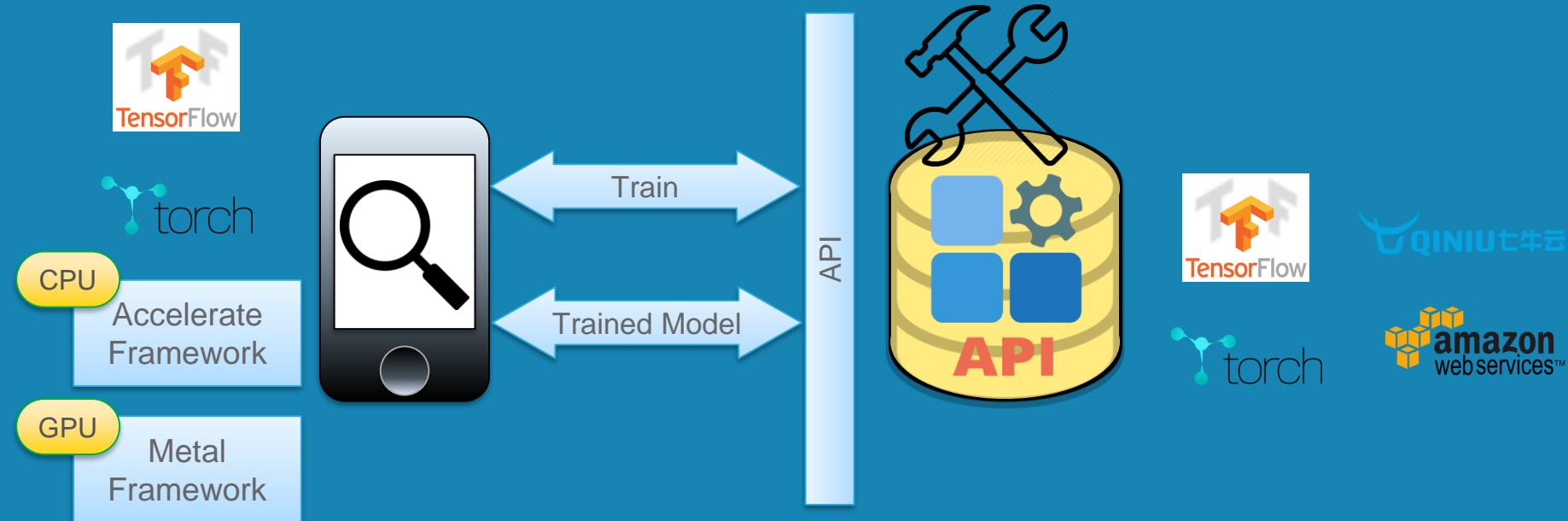
```
op@proxy:~ (ssh) 361 ~ (zsh) 362 ~ (zsh) 363 ~ (zsh) 364 ~ (zsh) 365
1 [ 0.0% 4 [ 0.0% 7 [ 1.3% 10 [ 0.0%
2 [ 0.0% 5 [ 0.7% 8 [ 0.0% 11 [ 0.0%
3 [ 0.0% 6 [ 0.0% 9 [ 0.0% 12 [ 0.0%
Mem[1 1 1] 1.11G/126G Tasks: 106, 278 thr; 1 running
Swp[ 0K/128G Load average: 0.00 0.00 0.00
Uptime: 2 days, 23:19:15

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
8678 op 20 0 29172 4032 3240 R 0.7 0.0 0:00.24 htop
2209 op 20 0 1353M 173M 72976 S 0.7 0.1 22:15.65 compiz
1449 op 20 0 886M 45412 27208 S 0.0 0.0 8:33.03 /usr/bin/dockerd -H fd://
2859 root 20 0 617M 15180 6988 S 0.0 0.0 0:14.83 docker-containerd -l unix:
1512 root 20 0 886M 45412 27208 S 0.0 0.0 1:24.93 /usr/bin/dockerd -H fd://
1513 root 20 0 886M 45412 27208 S 0.0 0.0 0:27.80 /usr/bin/dockerd -H fd://
1531 root 20 0 617M 15180 6988 S 0.0 0.0 4:42.41 docker-containerd -l unix:
2863 root 20 0 617M 15180 6988 S 0.0 0.0 0:16.64 docker-containerd -l unix:
1123 root 20 0 232M 58072 40140 S 0.0 0.0 3:01.21 /usr/lib/xorg/Xorg -core :
1 root 20 0 118M 5816 3992 S 0.0 0.0 0:02.89 /sbin/init splash
280 root 20 0 37444 10072 9640 S 0.0 0.0 0:01.29 /lib/systemd/systemd-journ
```

```
op@proxy:~/tensorflow$ bazel-bin/tensorflow/cc/tutorials_example_trainer --use_gpu
2017-06-27 16:31:08.990991: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
2017-06-27 16:31:08.991106: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
2017-06-27 16:31:08.991131: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-06-27 16:31:09.297409: I tensorflow/core/common_runtime/gpu/gpu_device.cc:940] Found device 0 with properties:
name: GeForce GTX 1080 Ti
major: 6 minor: 1 memoryClockRate (GHz) 1.683
pciBusID 0000:03:00.0
Total memory: 10.90GiB
Free memory: 10.69GiB
2017-06-27 16:31:09.297444: I tensorflow/core/common_runtime/gpu/gpu_device.cc:961] DMA: 0
2017-06-27 16:31:09.297452: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0: Y
2017-06-27 16:31:09.297466: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:03:00.0)
```

如何动手玩转ML - 在移动设备上使用训练好的模型

以iOS系统为例



如何动手玩转ML – 移动端训练的优缺点

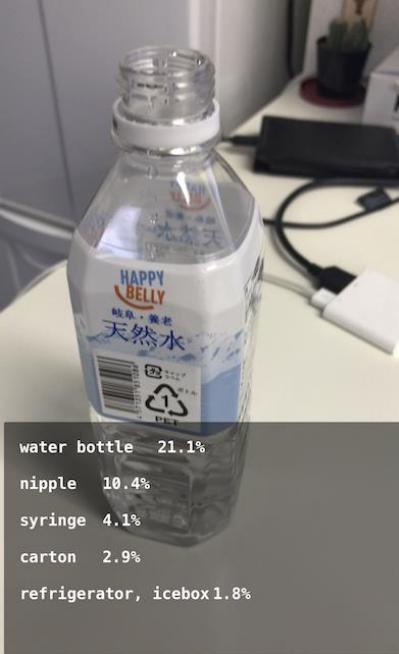
优点

- 可以自定义自己需要的模型
- 可以使用很多开源的模型
- 数据完全可控
- **可以做离线任务**

缺点

- 需要掌握ML引擎及工具
- 还得会Python
- 需要自己管理服务器资源占用
- **需要将复杂模型移动到移动端**
- **给模型优化带来很多限制**

< Back Metal CNN: Image Recognition



```
ViewController+alert.swift
Samples
LivePhotoCapture
UserNotification
SpeechRecognition
Looper
PropertyAnimator
MetalCNN
Common
  LivePhotoCapture
  UserNotification
  SpeechRecognition
  Looper
  PropertyAnimator
  MetalCNN
    Common
      SlimMPSCNN.swift
Basic
ImageRecognition
  InceptionNet.swift
  InceptionNet+result.swift
  MetalImageRecognitionViewController.swift
  MetalImageRecognition.storyboard
AttributedSpeech
New Image Filters
Audio Fade-in/out
StickerPack
PersistentContainer
TabBadge
New Fonts
PreviewInteraction
Proactive
Haptic
Resources
  plists
    iOS10AddedFonts.plist
network_params
  batch_normalized_binaries
    bias_conv.dat
    bias_conv_1.dat
    bias_conv_2.dat
    bias_conv_3.dat
  
```

```
//TODO: need to change to tensorflow files
// these are labels corresponding to output classes, these are defined by ImageNet2012 dataset which has 1000 classes
// the network given by tensorflow outputs 1008 class probabilities however as we have no labels for index 1001-1007, we
// have classes from original ImageNet2012
let labels = [
    "",
    "kit fox, Vulpes macrotis",
    "English setter",
    "Siberian husky",
    "Australian terrier",
    "English springer, English springer spaniel",
    "grey whale, gray whale, devilfish, Eschrichtius gibbosus, Eschrichtius robustus",
    "lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens",
    "Egyptian cat",
    "ibex, Capra ibex",
    "Persian cat",
    "cougar, puma, catamount, mountain lion, painter, panther, Felis concolor",
    "gazelle",
    "porcupine, hedgehog",
    "sea lion",
    " 
```

```
/*
This function gets a commandBuffer and encodes layers in it. It follows that by committing the commandBuffer
and getting labels
*/
func runNetwork() {
    let startTime = CACurrentMediaTime()

    // to deliver optimal performance we leave some resources used in MPSCNN to be released at next call of
    // autoreleasepool,
    // so the user can decide the appropriate time to release this
    autoreleasepool {
        // encoding command buffer
        let commandBuffer = commandQueue.makeCommandBuffer()

        // encode all layers of network on present commandBuffer, pass in the input image MTLTexture
        inception3Net.forward(commandBuffer: commandBuffer, sourceTexture: sourceTexture)

        // commit the commandBuffer and wait for completion on CPU
        commandBuffer.commit()
        commandBuffer.waitUntilCompleted()

        // display top-5 predictions for what the object should be labelled
        var resultStr = ""
        inception3Net.getResults().forEach({ (label, prob) in
            resultStr = resultStr + label + "\t" + String(format: "%.1f", prob * 100) + "%\n\n"
        })

        DispatchQueue.main.async {
            self.predictLabel.text = resultStr
        }
    }

    let endTime = CACurrentMediaTime()
    print("Running Time: \(endTime - startTime) [sec]")
}
```

The screenshot shows a PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar displays the project structure under the root directory `models`. It includes sub-directories like `tutorials`, `image`, and `im2txt`, along with various Python files and configuration files such as `mnist.py`, `mn.py`, `README.md`, and `CONTRIBUTING.md`.
- Code Editor:** The main window shows the file `classify_image.py` with the following content:

```
run_inference_o... with tf.Session... for node_id in ...
141     image_data = tf.gfile.FastGFile(image, 'rb').read()
142
143     # Creates graph from saved GraphDef.
144     create_graph()
145
146     with tf.Session() as sess:
147         # Some useful tensors:
148         # 'softmax:0': A tensor containing the normalized prediction across
149         #   1000 labels.
150         # 'pool_3:0': A tensor containing the next-to-last layer containing 2048
151         #   float description of the image.
152         # 'DecodeJpeg/contents:0': A tensor containing a string providing JPEG
153         #   encoding of the image.
154         # Runs the softmax tensor by feeding the image_data as input to the graph.
155         softmax_tensor = sess.graph.get_tensor_by_name('softmax:0')
156         predictions = sess.run(softmax_tensor,
157                               {'DecodeJpeg/contents:0': image_data})
158         predictions = np.squeeze(predictions)
159
160     # Creates node ID --> English string lookup.
161     node_lookup = NodeLookup()
162
163     top_k = predictions.argsort()[-FLAGS.num_top_predictions:][::-1]
164     for node_id in top_k:
165         human_string = node_lookup.id_to_string(node_id)
166         score = predictions[node_id]
167         print('%s (score = %.5f)' % (human_string, score))
168
169
170     def maybe_download_and_extract():
171         """Download and extract model tar file."""
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
226
227
228
229
229
230
231
232
233
234
235
235
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
```

```
op@mai:~/models/tutorials/image/imagenet$ python3 classify_image.py
>> Downloading inception-2015-12-05.tgz 100.0%
Successfully downloaded inception-2015-12-05.tgz 88931400 bytes.
2017-07-02 12:31:09.255855: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
2017-07-02 12:31:09.255896: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
2017-07-02 12:31:09.255904: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-07-02 12:31:09.463274: I tensorflow/core/common_runtime/gpu/gpu_device.cc:940] Found device 0 with properties
:
name: GeForce GTX 1080 Ti
major: 6 minor: 1 memoryClockRate (GHz) 1.683
pciBusID 0000:03:00.0
Total memory: 10.90GiB
Free memory: 10.52GiB
2017-07-02 12:31:09.463312: I tensorflow/core/common_runtime/gpu/gpu_device.cc:961] DMA: 0
2017-07-02 12:31:09.463320: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0: Y
2017-07-02 12:31:09.474841: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:03:00.0)
2017-07-02 12:31:09.985098: W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization().
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.89107)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00779)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score = 0.00296)
custard apple (score = 0.00147)
earthstar (score = 0.00117)
op@mai:~/models/tutorials/image/imagenet$
```

```
op@mai:~/models/tutorials/image/imagenet$ ll
total 181100
-rw-rw-r-- 1 op op      566  6月 27 18:45 BUILD
-rw-rw-r-- 1 op op 95673916 12月  5  2015 classify_image_graph_def.pb
-rw-rw-r-- 1 op op     7744  6月 27 18:45 classify_image.py
-rw-rw-r-- 1 op op    2683 12月   1  2015 cropped_panda.jpg
-rw-rw-r-- 1 op op  64986 11月 19  2015 imagenet_2012_challenge_label_map_proto.pbtxt
-rw-rw-r-- 1 op op 741401 11月 19  2015 imagenet_synset_to_human_label_map.txt
-rw-rw-r-- 1 op op 88931400 12月  5  2015 inception-2015-12-05.tgz
-rw-rw-r-- 1 op op   11416 12月  5  2015 LICENSE
op@mai:~/models/tutorials/image/imagenet$ less imagenet_2012_challenge_label_map_proto.pbtxt
op@mai:~/models/tutorials/image/imagenet$ less imagenet_synset_to_human_label_map.txt
op@mai:~/models/tutorials/image/imagenet$
```

```
# -*- protobuf -*-
# LabelMap from ImageNet 2012 full data set UID to int32 target class.
entry {
  target_class: 449
  target_class_string: "n01440764"
}
entry {
  target_class: 450
  target_class_string: "n01443537"
}
entry {
  target_class: 442
  target_class_string: "n01484850"
}
entry {
  target_class: 443
  target_class_string: "n01491361"
}
```

n00004475	organism, being
n00005787	benthos
n00006024	heterotroph
n00006484	cell
n00007846	person, individual, someone, somebody, mortal, soul
n00015388	animal, animate being, beast, brute, creature, fauna
n00017222	plant, flora, plant life
n00021265	food, nutrient
n00021939	artifact, artefact
n00120010	hop
n00141669	check-in
n00288000	dressage
n00288190	curvet, vaulting
n00288384	piaffe
n00324978	funambulism, tightrope walking
n00326094	rock climbing
n00433458	contact sport

移动设备上ML的训练和使用的未来会是什么样的？

- 希望模型从服务器端到移动端更方便
- 希望Apple和Google提供更多的算法及训练的API
- 做到模型的可拔插
- 希望能有更多的Swift/Java方面的ML工具出现

参考

Machine Learning APIs:

Google Prediction: <https://cloud.google.com/prediction>

Google Natural Language: <https://cloud.google.com/natural-language>

Microsoft Cognitive Services: <https://www.microsoft.com/cognitive-services>

Amazon ML: <https://aws.amazon.com/documentation/machine-learning>

IBM Watson: <https://www.ibm.com/watson/developercloud>

Open Source Model:

Tensor Flow Models: <https://github.com/tensorflow/models>

FaceNet for TensorFlow: <https://github.com/davidsandberg/facenet>

参考

Machine Learning Frameworks:

Torch: <http://torch.ch>

TensorFlow: <https://www.tensorflow.org>

Caffe: <https://github.com/BVLC/caffe>

Awesome Machine Learning resources: <https://github.com/josephmisiti/awesome-machine-learning>

Hosting:

Amazon Web Services: <https://aws.amazon.com>

Amazon Deep Learning AMI - Ubuntu Edition <https://aws.amazon.com/marketplace/pp/B06VSPXKDX>

Digital Ocean <https://www.digitalocean.com>

参考

Machine Learning Frameworks on iOS:

Torch: <https://github.com/clementfarabet/torch-ios>

TensorFlow: https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/ios_examples

<http://www.mattrajca.com/2016/11/25/getting-started-with-deep-mnist-and-tensorflow-on-ios.html>

Caffe: <https://github.com/noradaiko/caffe-ios-sample>

Using Metal Performance Shaders with a TensorFlow trained model:

<https://developer.apple.com/library/content/samplecode/MPSCNNHelloWorld>

Neural Networks and Accelerate: <https://developer.apple.com/videos/play/wwdc2016/715>

BNNS in Accelerate: <https://developer.apple.com/reference/accelerate/bnns>

List of ML resources for iOS: https://github.com/alexsosn/iOS_ML

参考

PPT

https://www.slideshare.net/keithmo/machine-learning-for-ios-developers?qid=523d6c19-52c1-48b2-9b2b-120cda80f3f0&v=&b=&from_search=3

Open Source:

Face Entry Example:

<https://github.com/keefmoon/faceentry>

提问 回答

Thanks!

讨论：

机器学习场景及训练模式优化

如何体现人工智能的优势