Myna：
Context Awareness Framework On Smart Devices

主讲人： TalkingData数据科学家 王小辉

Myna: Context Awareness Framework On Smart Devices

主讲人：王小辉 @ TalkingData

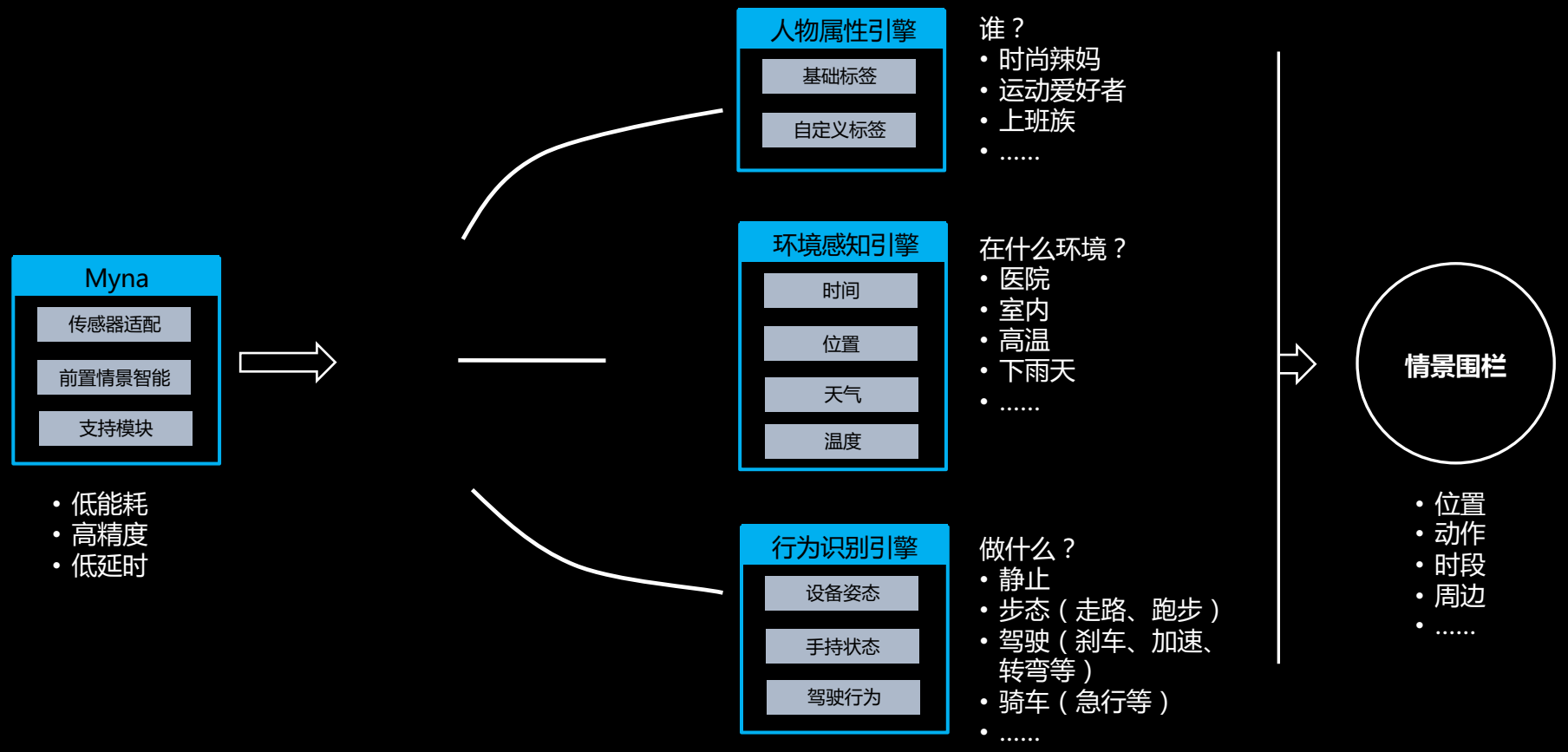
Agenda

- 什么是情景感知
- 情景感知的技术基础
- TalkingData Myna

什么是情景感知

什么人	什么时候	做什么	意图
上班族	早上 8 点，街道	走路	上班路上
外地人	下午 3 点，公园	走路	游览途中
运动狂人	傍晚 8 点，公园	跑步	健身
年轻女性	周末，街道	走路	逛街购物

什么是情景感知


Myna
传感器适配
前置情景智能
支持模块
• 低能耗
• 高精度
• 低延时
人物属性引擎
基础标签
自定义标签
谁？
• 时尚辣妈
• 运动爱好者
• 上班族
• ……
环境感知引擎
时间
位置
天气
温度
在什么环境？
• 医院
• 室内
• 高温
• 下雨天
• ……
行为识别引擎
设备姿态
手持状态
驾驶行为
做什么？
• 静止
• 步态（走路、跑步）
• 驾驶（刹车、加速、转弯等）
• 骑车（急行等）
• ……
情景围栏
• 位置
• 动作
• 时段
• 周边
• ……

什么是情景感知

情景感知的应用场景：

- UBI：驾驶行为识别在车险行业的应用
- Real Time & Right Time 基于情景感知结果的用户触达
- 室内定位和导航
- 应用反作弊
- 异常行为检测
- 人群画像
- Biometric & individual identification

情景感知的技术基础

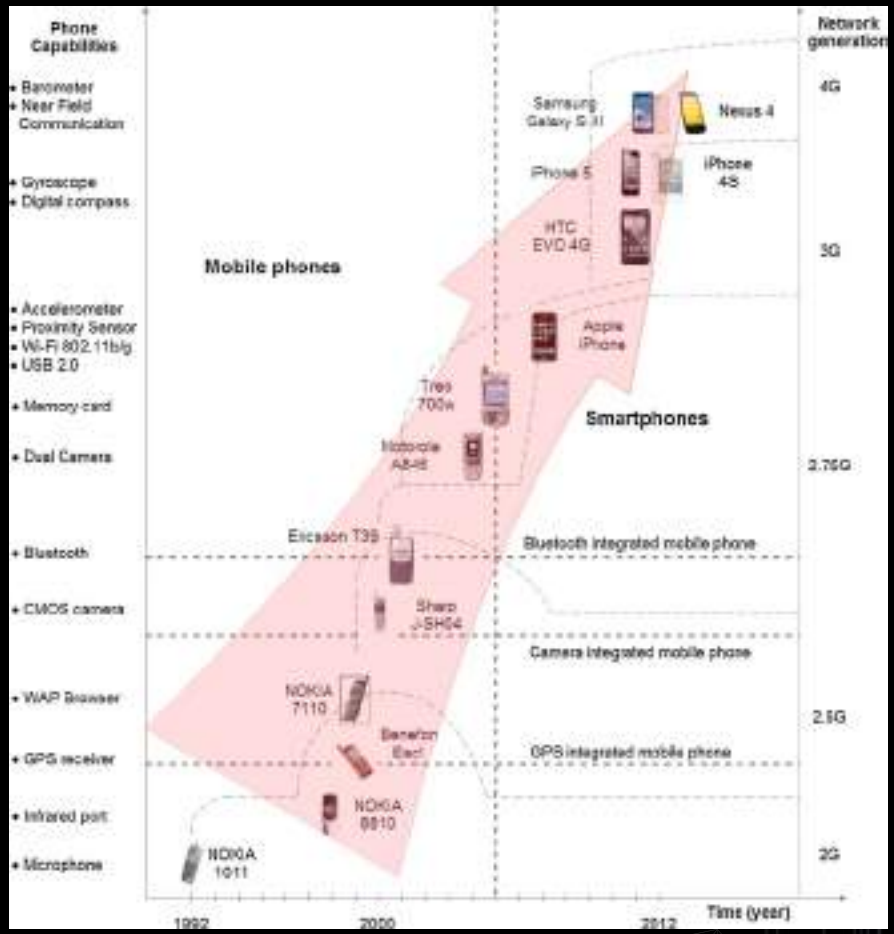
为什么传感器在智能移动设备上的发展是情景感知的基础？

趋势：

- 更多传感器类型
- 更高质量的传感器
- 更高频率的数据更新频率

挑战

- 智能应用的反应速度意味着更高频率的数据和更高的耗能
- 不同机型搭载的不同质量的传感器会影响模型泛化能力



情景感知的技术基础

iOS 和 Android 平台从系统层面对情景感知的支持：

iOS

- 2013, iBeacon geo-fencing
- 2013, CoreMotion in iOS7
- 2015, ResearchKit, Apple Watch
- 2016, Differential privacy
- 2017, CoreML & ARKit
- …

Android

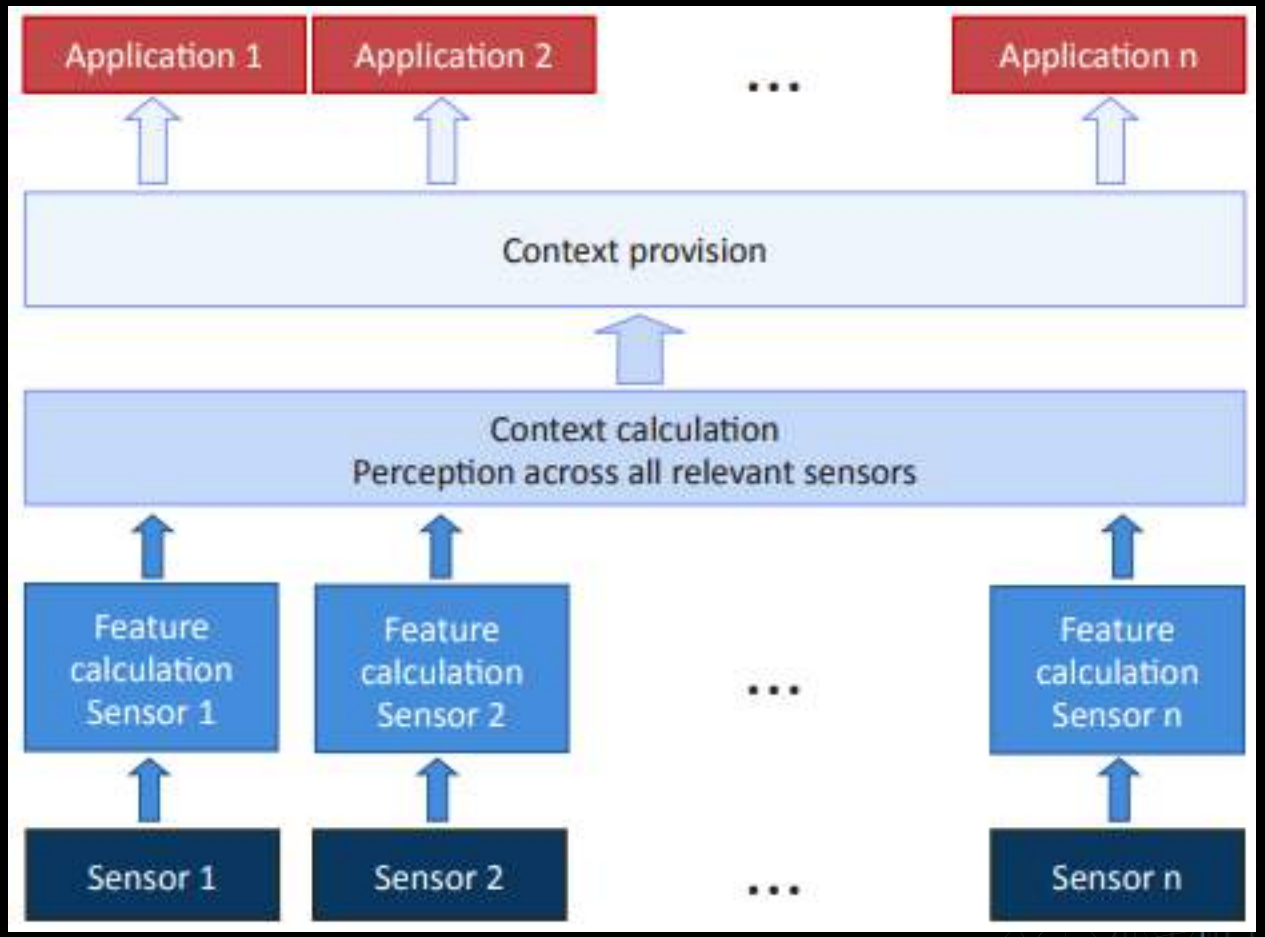
- 2014, DetectedActivity in Android 4.4
- 2015, Eddystone
- 2015, Power saving strategy based on recognized activities.
- 2016, Awareness API
- 2017, Android 8 and TensorFlow Lite
- …

情景感知的技术基础

如何实现情景感知？

基于传感器数据进行行为识别时，我们将做下面的事：

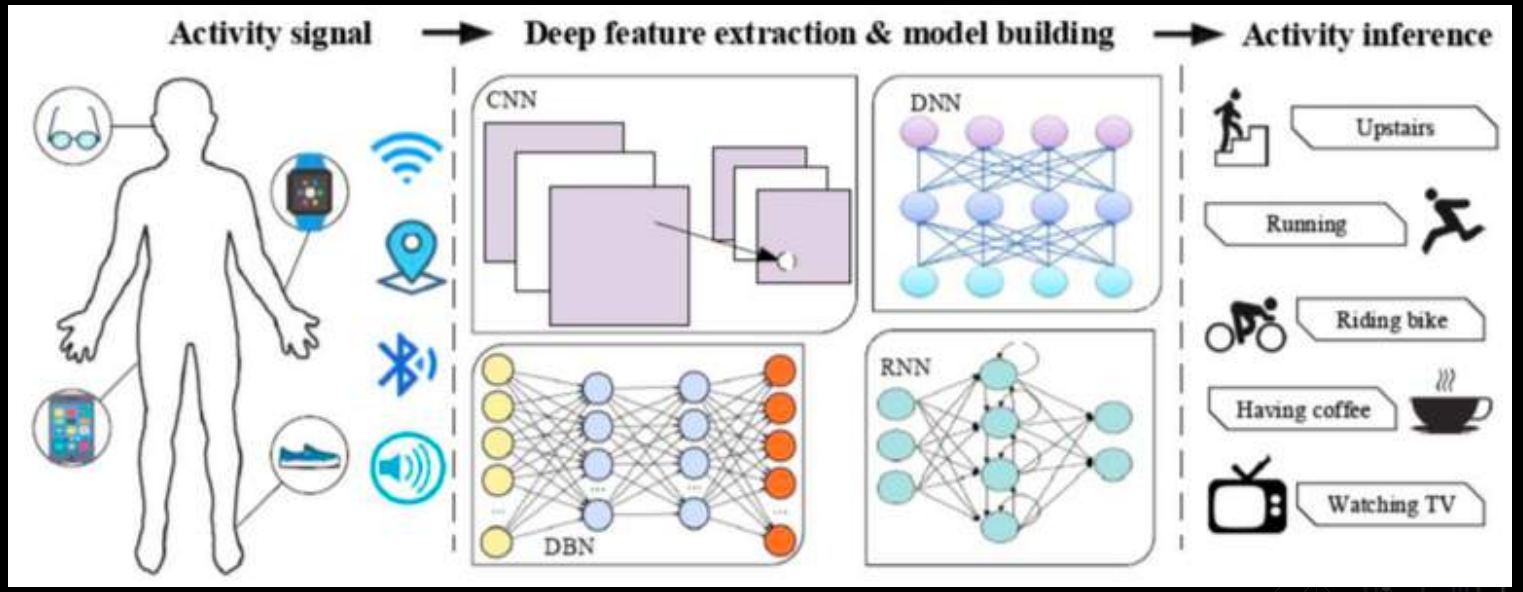
1. 理解移动平台传感器数据及其坐标系统。
2. 采集训练和测试数据集。
3. 数据准备
4. 特征抽取和选择
5. 模型训练和评估
6. 实时数据测试。



情景感知的技术基础

深度学习？

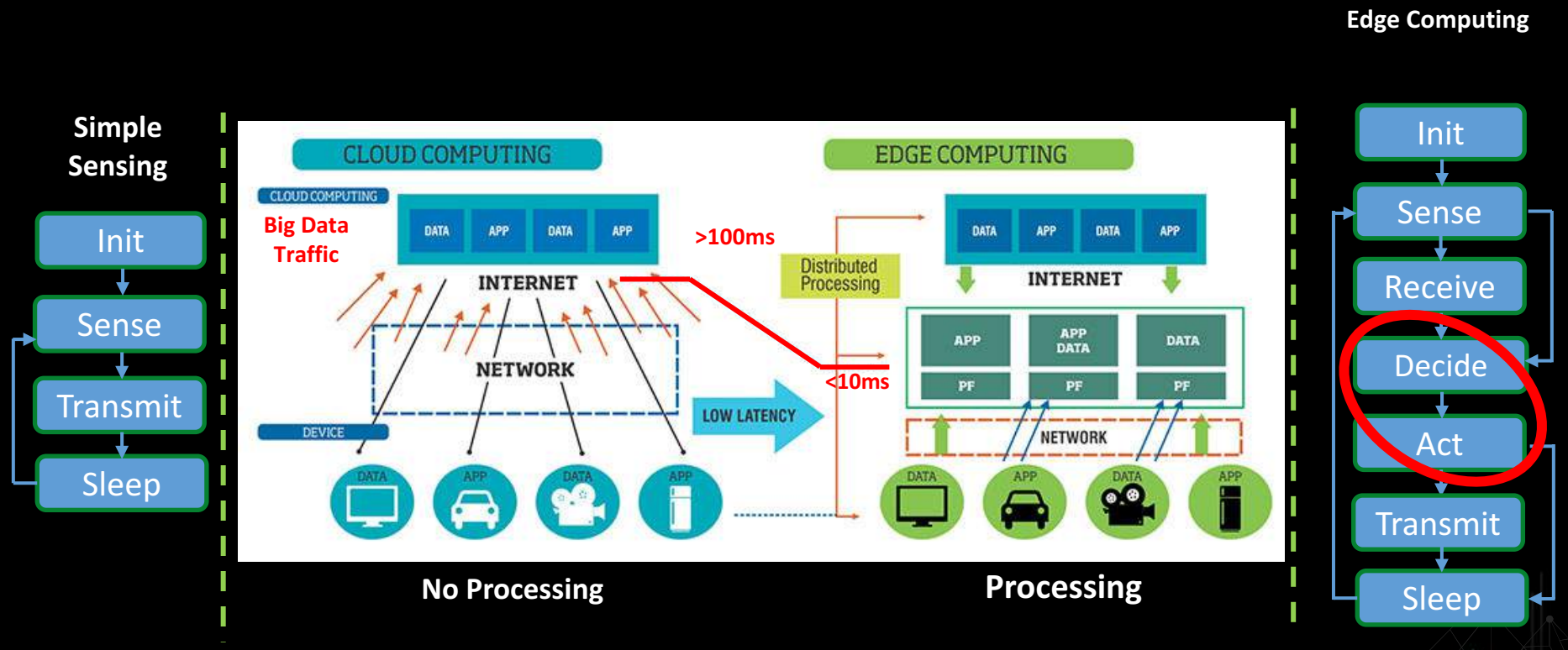
将特征提取与模型构建的步骤合二为一，自动提取数 据中的深度特征，提升了模型的泛化能力，并且也能够提取出模型更加深度的特征。



情景感知的技术基础

为什么在移动端？

低成本
低延时
高可靠性



情景感知的技术基础

目前前沿的深度学习方案？

Deepsense： **a unified deep learning framework for time-series mobile sensing data processing**

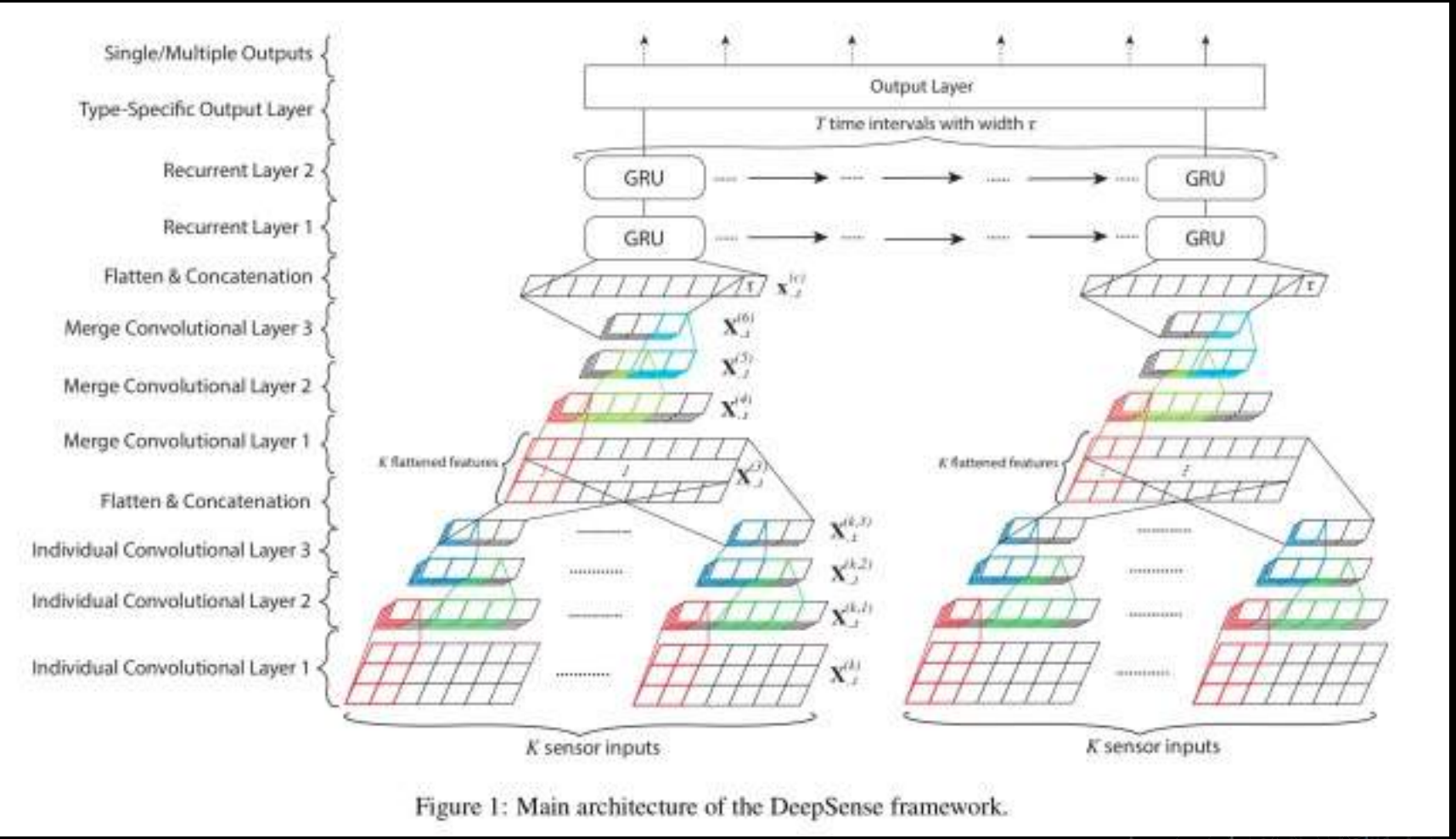


Figure 1: Main architecture of the DeepSense framework.

情景感知的技术基础

为什么现在重视？

- 2017 年 3 月 ARM 提供了支持 Context-A 系列 CPU 和 Mali 系列 GPU 的 Compute Library，让机器学习和深度学习算法在 ARM 平台更高效地运行，并提出了面向 AI 的新架构 DynamIQ 技术。
- 2017 年 8 月高通对外发布了支持 Caffe/Caffe2 和 TensorFlow 的 Neural Processing Engine SDK，让深度学习可以利用 GPU 和 DSP 的计算能力。
- 2017 I/O 大会，Google 宣布为 Android 平台推出 TensorFlow Lite。
- 2017 WWDC 上，Apple 发布了支持 Caffe、TensorFlow 等深度学习框架，SVM、XGBoost 以及 sklearn 等机器学习模型的 CoreML，其内部实现可以利用到 Metal 2 硬件。
- 2017 年 9 月，华为发布了麒麟 970，内置中科院寒武纪-1A NPU。
- Google 开源了转为移动端优化的 MobileNets 模型；Face++ 提出了适合移动端的 ShuffleNet 模型。OpenCV 3.3 内置了支持多种深度学习框架的 DNN。

情景感知的技术基础

https://github.com/TalkingData/Myna

Myna

Android 平台情景感知框架

license Apache 2.0

Myna

Myna 项目中包含一个测试 Demo 工程：demo-myna, 将该工程和 Myna 项目本身导入到 Android Studio 中，就可以开始调试了。

Myna 提供了两套接口：

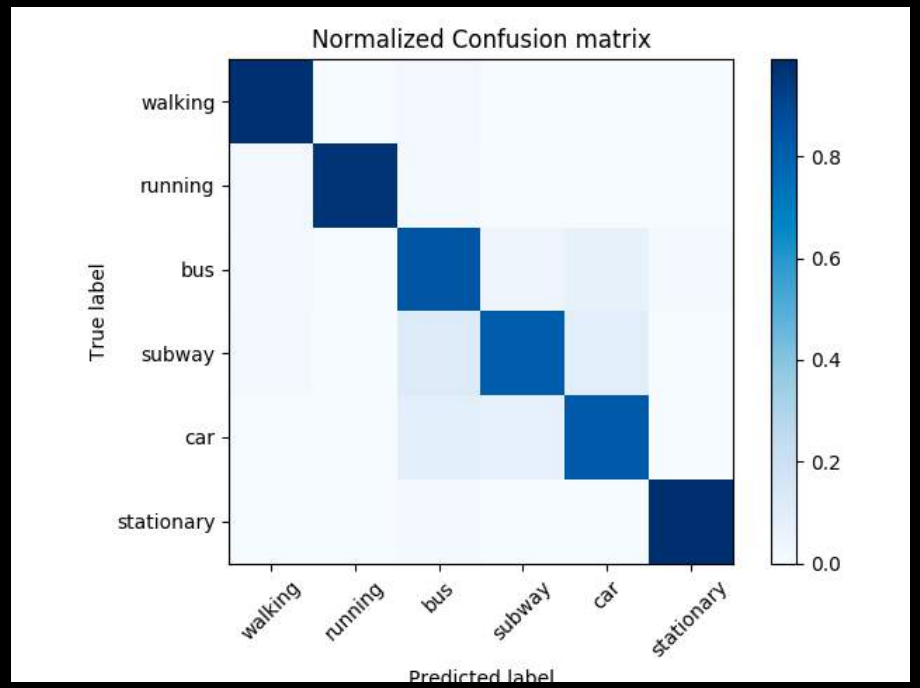
- 面向开发者的接口：开发者只需要简单的接口调用，就能在应用程序中获取实时识别的用户行为状态。
- 面向数据科学家的接口：数据科学家可以很方便地添加新的识别算法，在运行时调整订阅的传感器类型、采样频率和采样时长，而无需关心 Android 系统相关的传感器数据订阅细节。

TalkingData Myna

TalkingData Myna：Android 平台情景感知 Framework

目前支持识别的行为类型：

- 静止
- 走
- 跑
- 乘坐公交车
- 乘坐地铁
- 乘坐小汽车



Myna 提供了两套接口：

- 面向开发者的接口：开发者只需要简单的接口调用，就能在应用程序中获取实时识别的用户行为状态。
- 面向数据科学家的接口：数据科学家可以很方便地添加新的识别算法，在运行时调整订阅的传感器类型、采样频率和采样时长，而无需关心 Android 系统相关的传感器数据订阅细节。

TalkingData Myna

面向数据科学家的接口

DataScientistAPI 中定义了更多高级接口。MynaRecognizerInterface 类型中定义了对订阅的传感器类型进行控制的接口。

- 让数据科学家 Focus 在数据、特征工程和算法模型
- 数据科学家不需要了解 Android 系统的传感器开发细节
- 可配置项
  - 订阅的传感器类型
  - 采样频率
  - 采样时间窗口
  - 为同一组数据运行多个分类器

添加新的传感器类型订阅：

```
/**
 * Add a sensor into the chosen sensor list.
 * @param sensorType sensorType
 */
public synchronized void addSensorType(int sensorType)
```

移除已经订阅的传感器类型：

```
/**
 * Remove a sensor from the chosen sensor list.
 * @param sensorType The type of the sensor to be removed.
 */
public synchronized void removeSensorType(int sensorType)
```

设置采样的间隔时间（反映采样频率，单位毫秒）：

```
/**
 * Set sampling duration.
 * @param duration Sampling duration
 */
public void setSamplingDuration(int duration)
```

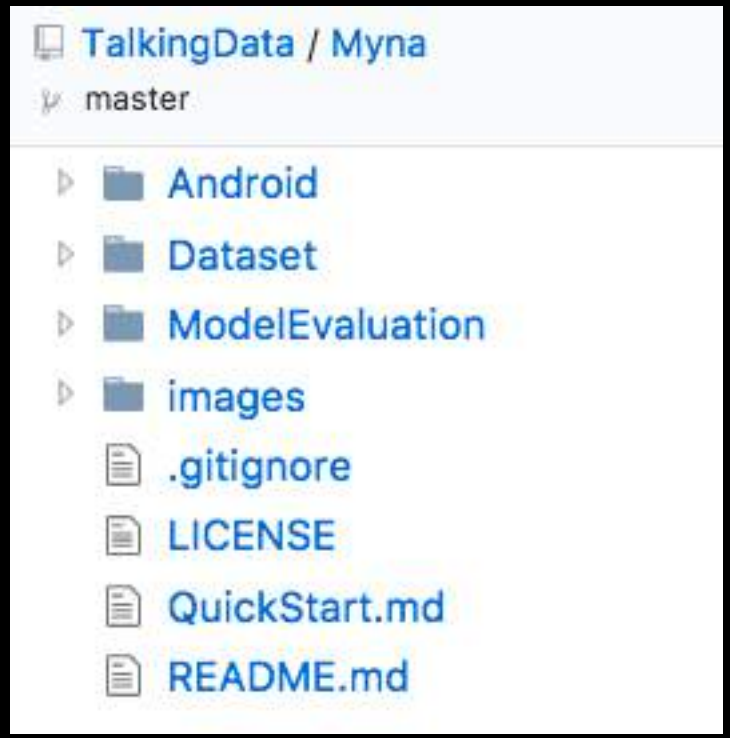
设置 batch size：

```
/**
 * Set total count of the data points for each recognition.
 * @param pointCount Total count of the data points.
 */
public void setSamplingPointCount
```

TalkingData Myna

TalkingData Myna 开源的内容包括那些？

- 训练和测试数据集
- 进行特征抽取的 Python 和 Java 代码
- 训练随机森林算法的 Java 代码
- 训练 XGBoost 模型的 Python 脚本
- 模型评估的 Python 脚本
- Demo 示例工程
- 详细接口 API 说明文档。



TalkingData Myna

训练数据哪里来？

我们评估了两个用户行为识别的公开数据集：

1. UCI：Human Activity Recognition Using Smartphones Dataset
2. WISDM：Wireless Sensor Data Mining Dataset
3. TalkingData Dataset

Statistics

Raw Time Series Data

- Number of examples: 1,098,207
- Number of attributes: 6
- Missing attribute values: None
- Class Distribution
  - Walking: 424,400 (38.6%)
  - Jogging: 342,177 (31.2%)
  - Upstairs: 122,869 (11.2%)
  - Downstairs: 100,427 (9.1%)
  - Sitting: 59,939 (5.5%)
  - Standing: 48,395 (4.4%)

Data Set Information:

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

TalkingData Myna

如何进行有效的训练数据采集？

用户处于相同的行为状态，不同的设备姿态会直接影响传感器数据在机身坐标下三个轴上的取值。

机身坐标系如右图所示，不难想到如果设备处于下面的姿态，即使用户一直处于同一种行为状态，传感器数据也是不一样的：

- 放在包中
- 装在裤兜
- 装在上衣口袋
- 那种手中随手臂自然摆动
- 使用中

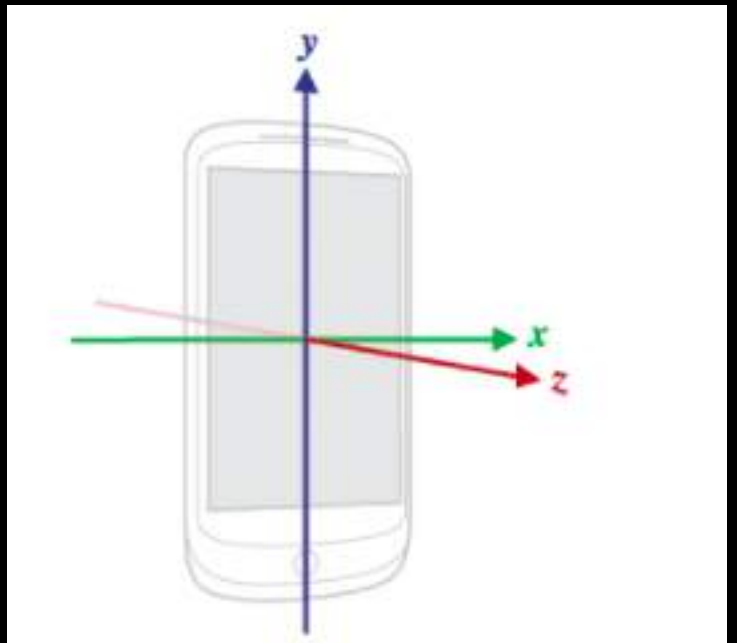
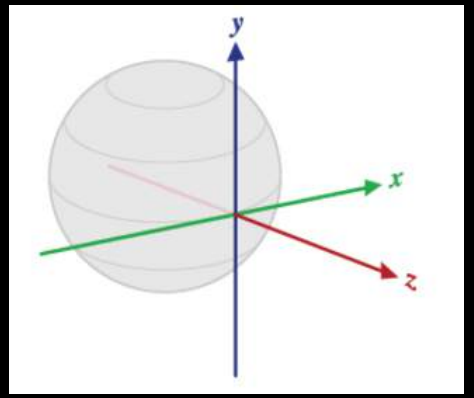


Figure 1. Coordinate system (relative to a device) that's used by the Sensor API.

TalkingData Myna

如何进行有效的训练数据采集？



通过磁场传感器和重力加速度传感器计算得到从机身坐标系到地球坐标系的旋转矩阵；然后利用该旋转矩阵将加速度转换到地球坐标系，摆脱同一个行为状态下不同机身姿态的影响。

getRotationMatrix

Added in API level 3

```
boolean getRotationMatrix (float[] R,
                float[] I,
                float[] gravity,
                float[] geomagnetic)
```

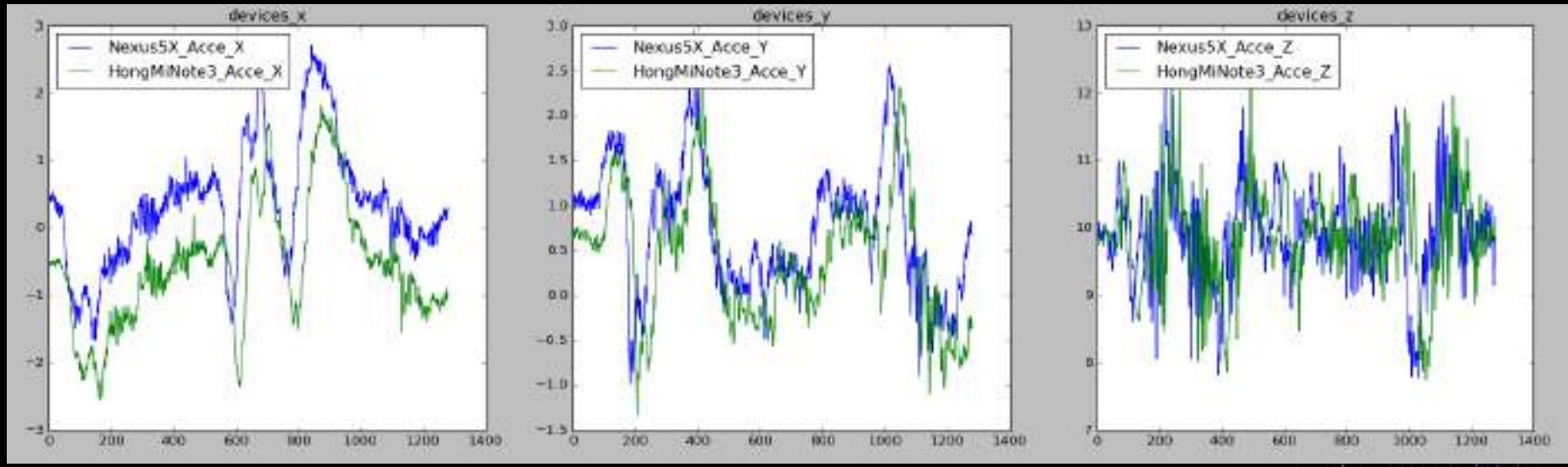
Computes the inclination matrix **I** as well as the rotation matrix **R** transforming a vector from the device coordinate system to the world's coordinate system which is defined as a direct orthonormal basis, where:

- X is defined as the vector product **Y.Z** (It is tangential to the ground at the device's current location and roughly points East).
- Y is tangential to the ground at the device's current location and points towards the magnetic North Pole.
- Z points towards the sky and is perpendicular to the ground.

TalkingData Myna

如何保证模型对不同机型的兼容性？

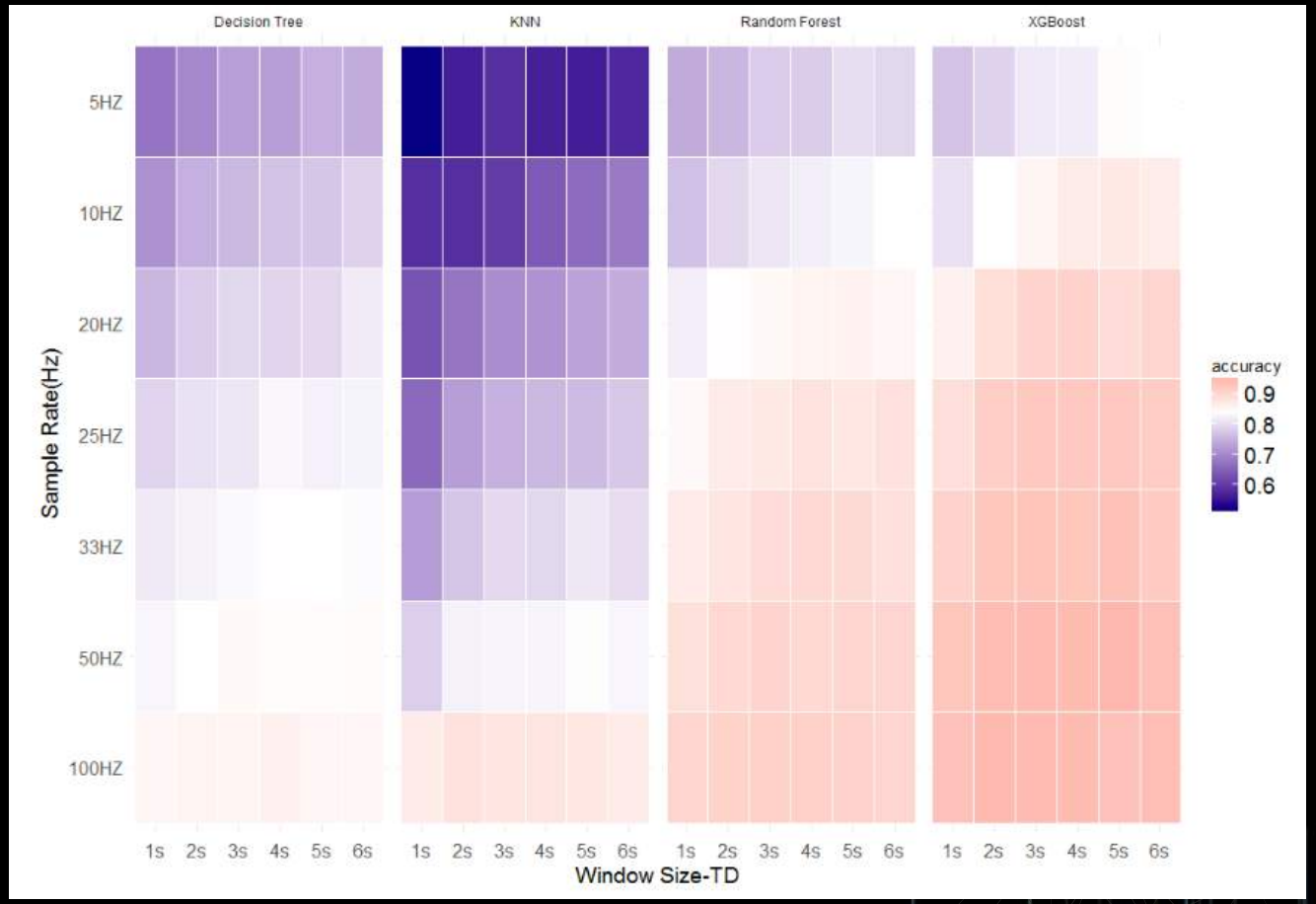
对 Android 平台来说，不同价位的设备对应不同质量的硬件，同时也意味着不同质量的传感器硬件平台。经过测试发现，尽快传感器数值在不同机型上有明显不同，但趋势一致。下图表示两个不同设备在完全相同的行为过程中采集到的数据。



TalkingData Myna

如何确定数据采集频率和时间窗口？

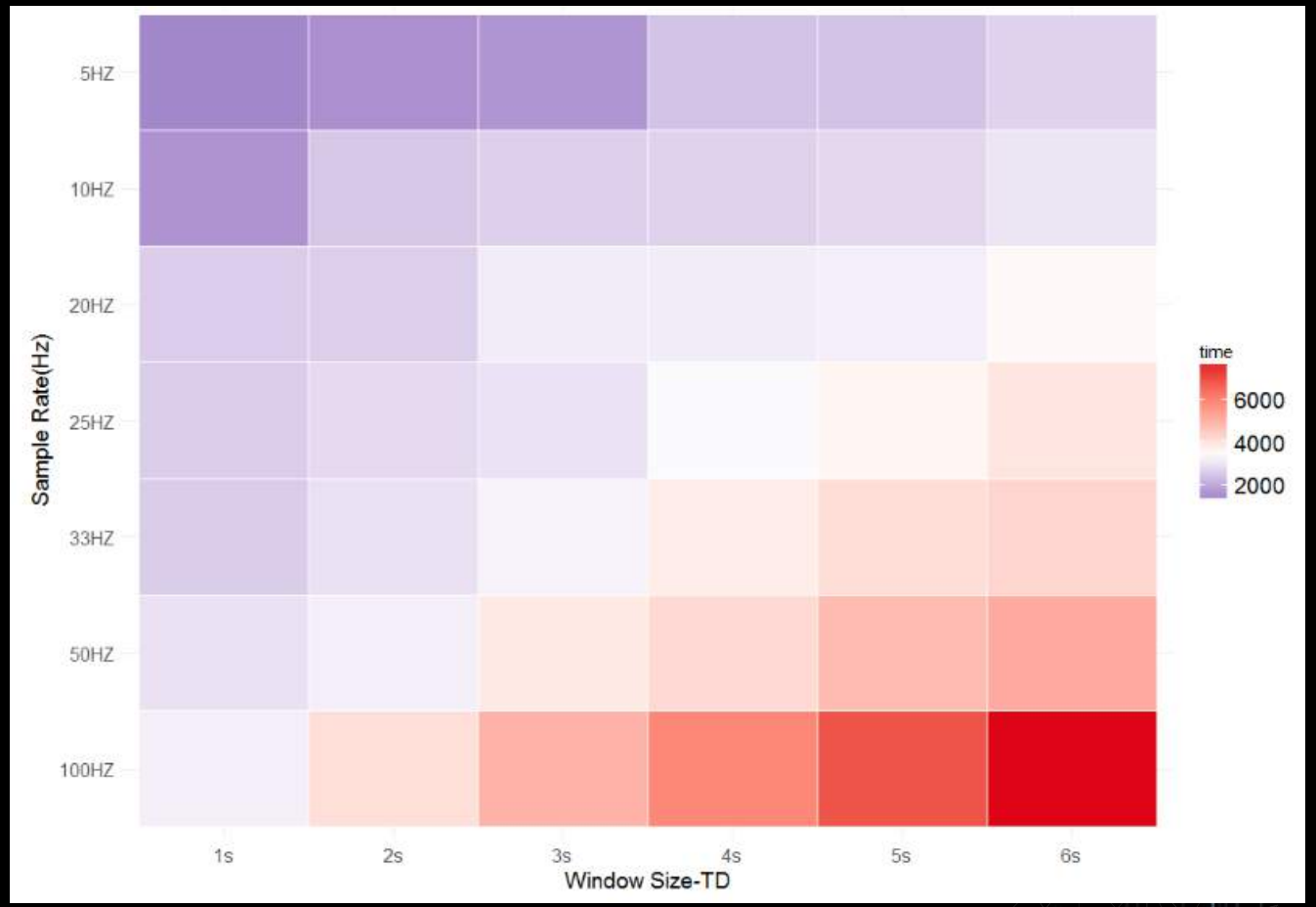
选择合适的采样频率和时间窗口长度，从右图的对比来看，很显然更长的时间窗口和更高的采样频率可以提高数据的质量，而且提升模型识别的准确率。



TalkingData Myna

如何平衡模型准确率和耗电量？

但是在移动平台，我们还要平衡性能，所以可以取折中方案：较长的时间窗口和降低的采样频率，保证一个时间窗口内的数据量，保证准确率。



TalkingData Myna

如何选择合适的特征维度？

传感器数据是典型的 Time Series Signals，对这类信号的特征抽取来说，时域特征和频域特征都是需要考虑的。

- SFFS (Sequential Forward Floating Search) 设定空的选定特征集 X 和备选特征集 S。在给定 X 下，S 中最显著的特征，指的是该特征加入 X 可以使得训练错误率最小。在每 次迭代中，S 中最显著的特征将移入 X，而 X 中移除后准确率改进的特征将移回 S。
- XGBoost Feature Importance：根据特征对正确识别的重要性做排名，选择最重要的维度。

特征抽取是计算密集型操作，也是比较耗能的操作，我们需要 trade off 模型准确率和计算复杂度。

TalkingData Myna

如何训练模型参数？

先根据经验为每个参数设定取值范围，然后使用 sklearn.model_selection 包中提供的 GridSearchCV 对每个 parameter 分别训练并验证，选择模型准确率最高的参数值。

```
def train_max_depth(lr):
    param_test1 = {
        'max_depth': range(3, 10, 1),
        'min_child_weight': range(1, 6, 1)
    }
    k_fold = StratifiedKFold(n_splits=10, shuffle=True, random_state=7)
    grid_search = GridSearchCV(
        estimator=XGBClassifier(
            objective='multi:softmax',
            learning_rate=lr,
            n_estimators=100,
            nthread=4,
            scale_pos_weight=1,
            seed=27),
        param_grid=param_test1, n_jobs=4, iid=False, cv=k_fold)
    grid_search.fit(train_x, train_y)
    grid_result = grid_search.fit(train_x, train_y)
    print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
    print_searching_details(grid_result)
    return grid_result.best_params_["max_depth"], grid_result.best_params_["min_child_weight"]
```

TalkingData Myna

如何在移动端进行 inference ?

- 按照设定的采样频率和时间窗口长度订阅选定传感器的数据。
- 抽取特征，对于乘坐不同交通方式的行为，去除噪声的效果反而不好，故不应用滤波算法。
- 使用预训练的模型对这些特征进行 inference，得到每种行为可能的 possibility 并输出。

```java
/**
 * Extract and select features from the raw sensor data points.
 * These data points are collected with certain sampling frequency and windows.
 * @param sensorData Raw sensor data points.
 * @return Extracted features.
 */
private double[] prepareFeatures(SensorData[] sensorData, final int sampleFreq, final int sampleCount) {

    double[] matrix = new double[SensorFeature.FEATURE_COUNT];
    Feature aFeature = new Feature();
    aFeature.extractFeatures(sensorData, sampleFreq, sampleCount);
    System.arraycopy(aFeature.getFeaturesAsArray(), 0, matrix, 0, SensorFeature.FEATURE_COUNT);
    return matrix;
}

/**
 * Recognize current human activity based on pre-defined rules.
 * @param sensorData Raw sensor data points.
 */
@Override
public double[] recognize(SensorData[] sensorData, final int sampleFreq, final int sampleCount) {
    features = prepareFeatures(sensorData, sampleFreq, sampleCount);
    return predict();
}

@Override
public double[] getCurrentFeatures() { return features; }

private double[] predict() {
    FVec vector = FVec.Transformer.fromArray(features, true);
    return predictor.predict(vector);
}
```

TalkingData Myna

如何导出 TensorFlow LSTM 行为识别模型？

```python
def optimize_model():
    from tensorflow.python.tools import freeze_graph
    from tensorflow.python.tools import optimize_for_inference_lib

    freeze_graph.freeze_graph(input_graph="graph.pb",
                              input_checkpoint="weights.ckpt",
                              output_node_names="output",
                              output_graph="frozen.pb",
                              input_binary=False,
                              restore_op_name="save/restore_all",
                              filename_tensor_name="save/Const:0",
                              clear_devices=True,
                              input_saver="",
                              initializer_nodes="")

    input_graph_def = tf.GraphDef()
    with tf.gfile.Open("frozen.pb", "r") as f:
        data = f.read()
        input_graph_def.ParseFromString(data)

    output_graph_def = optimize_for_inference_lib.optimize_for_inference(
        input_graph_def,
        ["input"],
        ["output"],
        tf.float32.as_datatype_enum)

    f = tf.gfile.FastGFile("optimized_frozen.pb", "w")
    f.write(output_graph_def.SerializeToString())
```

TalkingData Myna

如何在移动端加载运行 TensorFlow LSTM 行为识别模型？

```
private TensorFlowInferenceInterface inferenceInterface;
private static final String MODEL_FILE = "file:///android_asset/optimized_frozen_lstm.pb";
private static final String INPUT_NODE = "input";
private static final String[] OUTPUT_NODES = {"output"};
private static final String OUTPUT_NODE = "output";
private static final int INPUT_SIZE = 3 * 128;
private static final int OUTPUT_SIZE = 6;
```

```
public double[] recognize(SensorData[] sensorData, final int sampleFreq, final int sampleCount) {
    float[] input = prepareFeatures(sensorData, sampleCount);
    float[] result = new float[OUTPUT_SIZE];

    inferenceInterface.feed(INPUT_NODE,input,INPUT_SIZE);
    inferenceInterface.run(OUTPUT_NODES);
    inferenceInterface.fetch(OUTPUT_NODE,result);
    double[] doubleResult = new double[OUTPUT_SIZE];
    for(int i = 0; i < OUTPUT_SIZE; ++i){
        doubleResult[i] = (double)result[i];
    }
    return doubleResult;
}
```


T11
2017
THANKS
TalkingData