射頻系統深度學習

ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

概述

- 簡介
- 射頻系統中的深度學習
- Deepwave Digital技術
- 訊號檢測和分類示例
- GPU的即時DSP基準測試
- 總結

深度學習和射頻(RF)系統

深度學習方興未艾

網路

- 入侵偵測
- 威脅分類
- 面部識別
- 影像分析

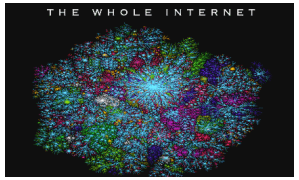
醫學

- 腫瘤檢測
- 醫學數據分析
- 診斷
- 藥物發現

自動化

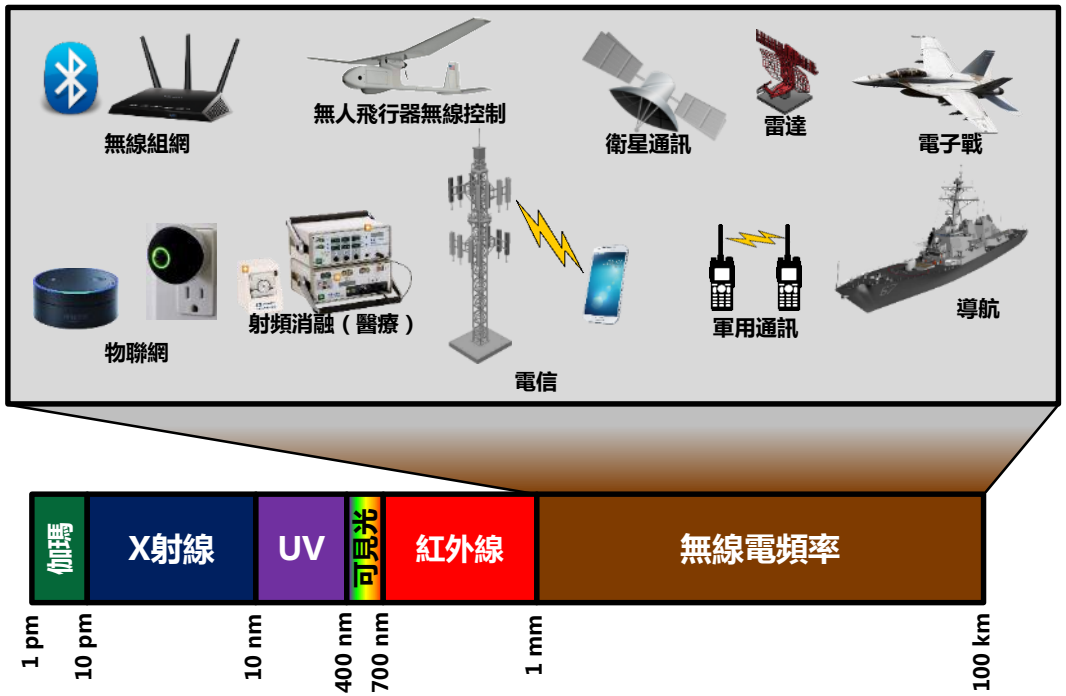
- 行人/障礙物檢測
- 導航
- 路牌識別
- 語音識別

互聯網



- 圖片分類
- 語音識別
- 語言翻譯
- 文件/資料庫搜索

射頻技術無處不在



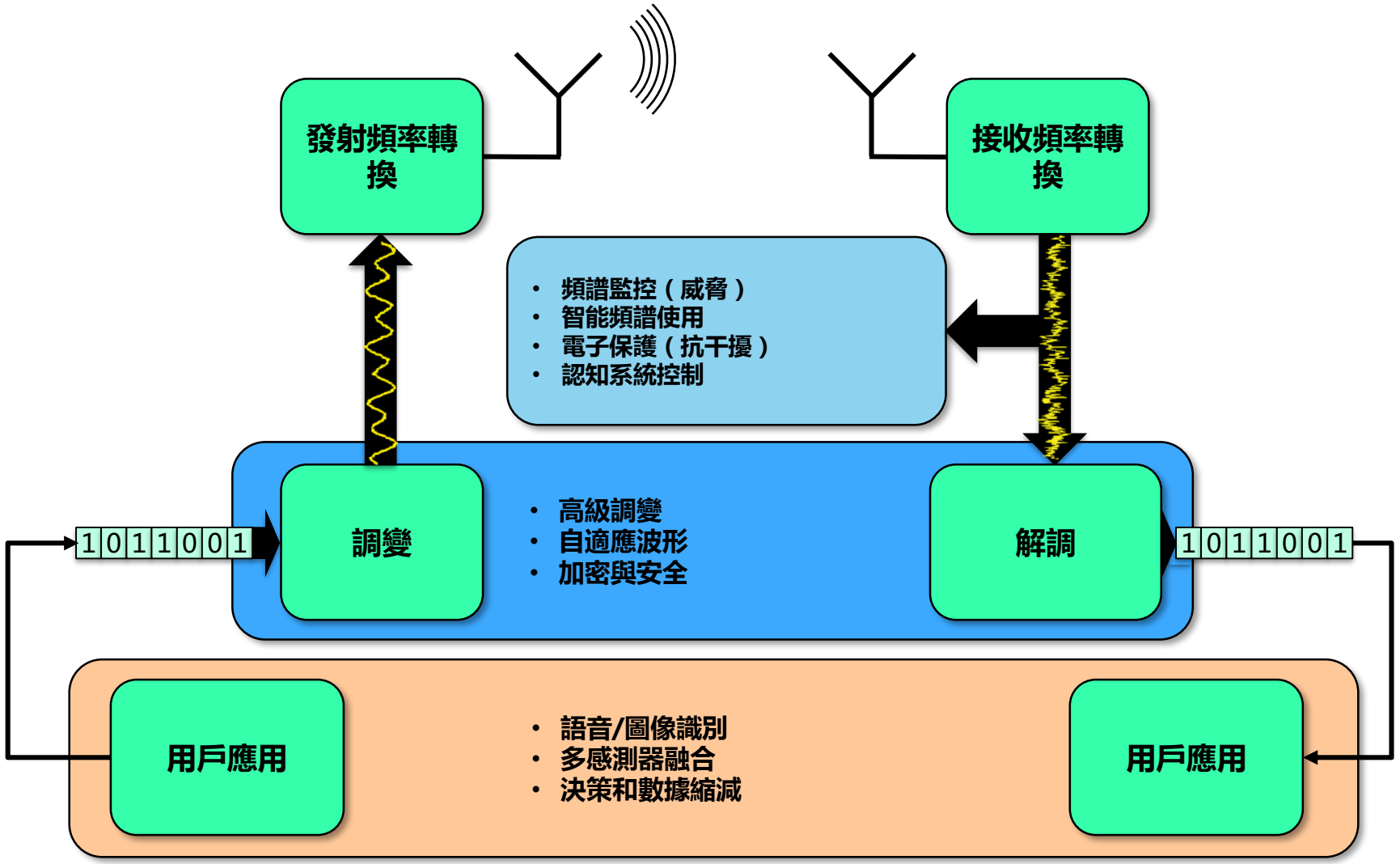
深度學習技術尚未對射頻系統產生重大影響

射頻系統中什麼地方使用深度學習

以頻譜/網路為中心的應用

以元件/基地台為中心的應用

以用戶應用為中心的應用



為什麼沒有解決?

頻寬限制

無法進行遠端處理

- AI需要大資料集
- 頻寬不足，無法發送到遠端資料中心

運算資源有限

現場

- 尚不存在整合AI運算處理器的RF系統

軟體複雜

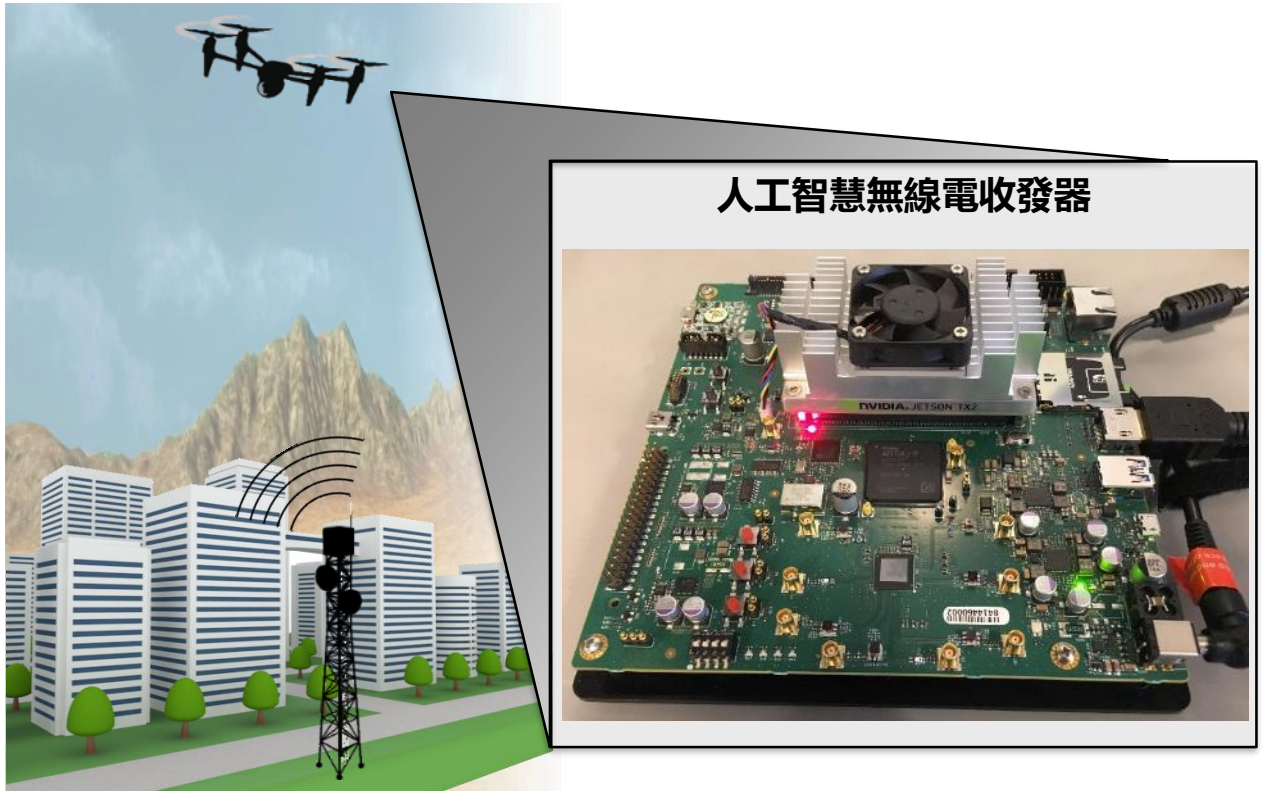
分別用於RF和AI

- 相互脫節的軟體
- 難以編程和理解

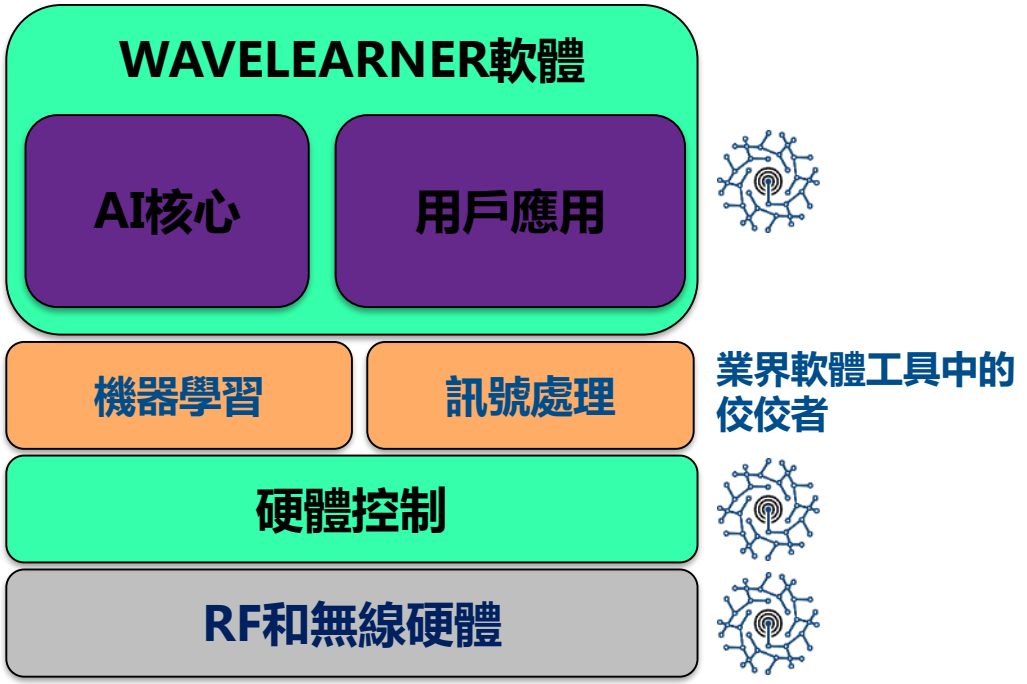
Deepwave解決方案和平台

方法 – 透過我們整合的硬體和軟體平台使AI在無線技術中得到廣泛採用

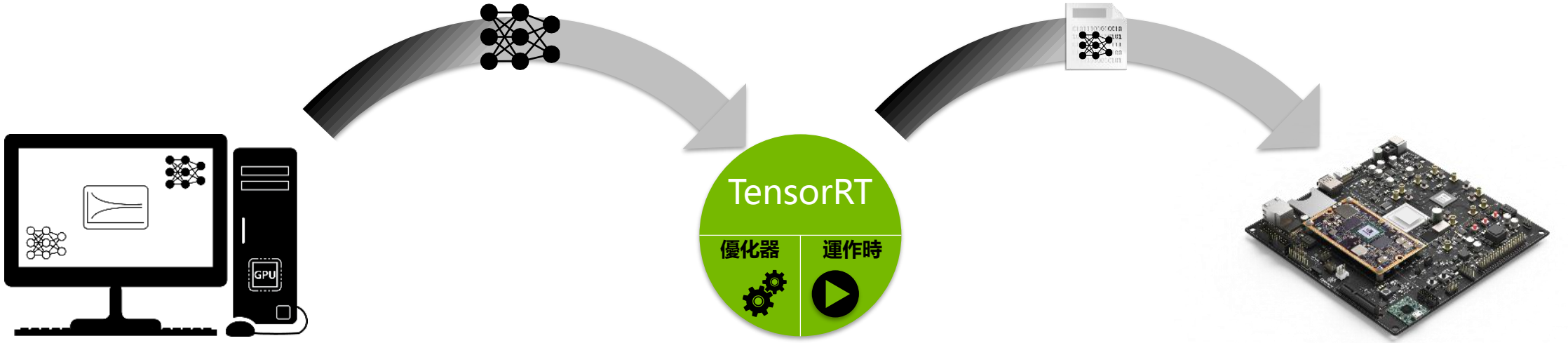
針對實際應用的硬體



易於編程的軟體



借助GR-Wavelearner實現邊緣推理



訓練神經網路

優化神經網路

佈署應用

TensorFlow PyTorch mxnet Cognitive Toolkit Caffe2 PaddlePaddle Chainer

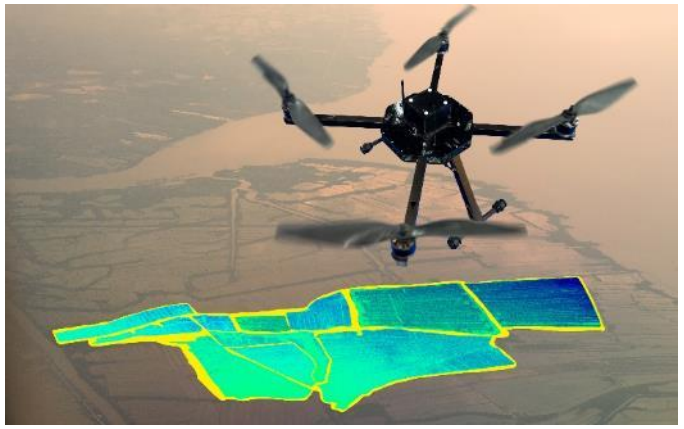
TensorRT NVIDIA

DEEPWAVE DIGITAL GNURadio

射頻系統中的深度學習

深度學習比較

圖像和影像



- 多通道(RGB)
- x、y空間依賴性
- 時間依賴性（視頻）

音訊和語言

- 單通道
- 頻率、相位、幅度
- 時間依賴性

系統和訊號

- 多個通道
- 頻率、相位、幅度
- 時間依賴性
- 複合數據(I/Q)
- 大頻寬
- 人因工程

現有深度學習有適用於系統和訊號的潛力

- **必須應對寬頻訊號和複雜的資料類型**

射頻系統中的深度學習硬體

	訓練	
	優點	**缺點**
CPU	• ML框架支援 • 功耗較低	• 比GPU慢 • 軟體架構較少
GPU	• ML框架支援 • 廣泛使用 • 高度並行/適應性強 • 良好的輸送量與功耗比	• 總功耗 • 需要高度並行的演算法
FPGA	使用不廣泛，（尚）不太適合	
ASIC	使用不廣泛，不太適合	

射頻系統中的深度學習硬體

	訓練		推理	
	優點	缺點	優點	缺點
CPU	• ML框架支持 • 功耗較低	• 比GPU慢 • 軟體架構較少	• 自我調整架構 • 軟體可程式設計 • 中等延遲	• 低並行度 • 即時頻寬有限 • 中等功率需求
GPU	• ML框架支持 • 廣泛使用 • 高度並行/適應性強 • 良好的輸送量與功耗比	• 總功耗 • 需要高度並行的演算法	• 自我調整架構 • 高即時頻寬 • 軟體可程式設計	• 中等功率需求 • 沒有很好地整合到RF中 • 延遲較高
FPGA	使用不廣泛，（尚）不太適合		• 高功效比 • 高即時頻寬 • 低延遲	• 開發/升級時間長 • 重程式設計能力有限 • 需要特殊專業知識
ASIC	使用不廣泛，不太適合		• 極高功效比 • 高即時頻寬 • 高度可靠 • 低延遲	• 極其昂貴 • 開發時間長 • 不能重新程式設計 • 需要特殊專業知識

射頻系統中的深度學習硬體

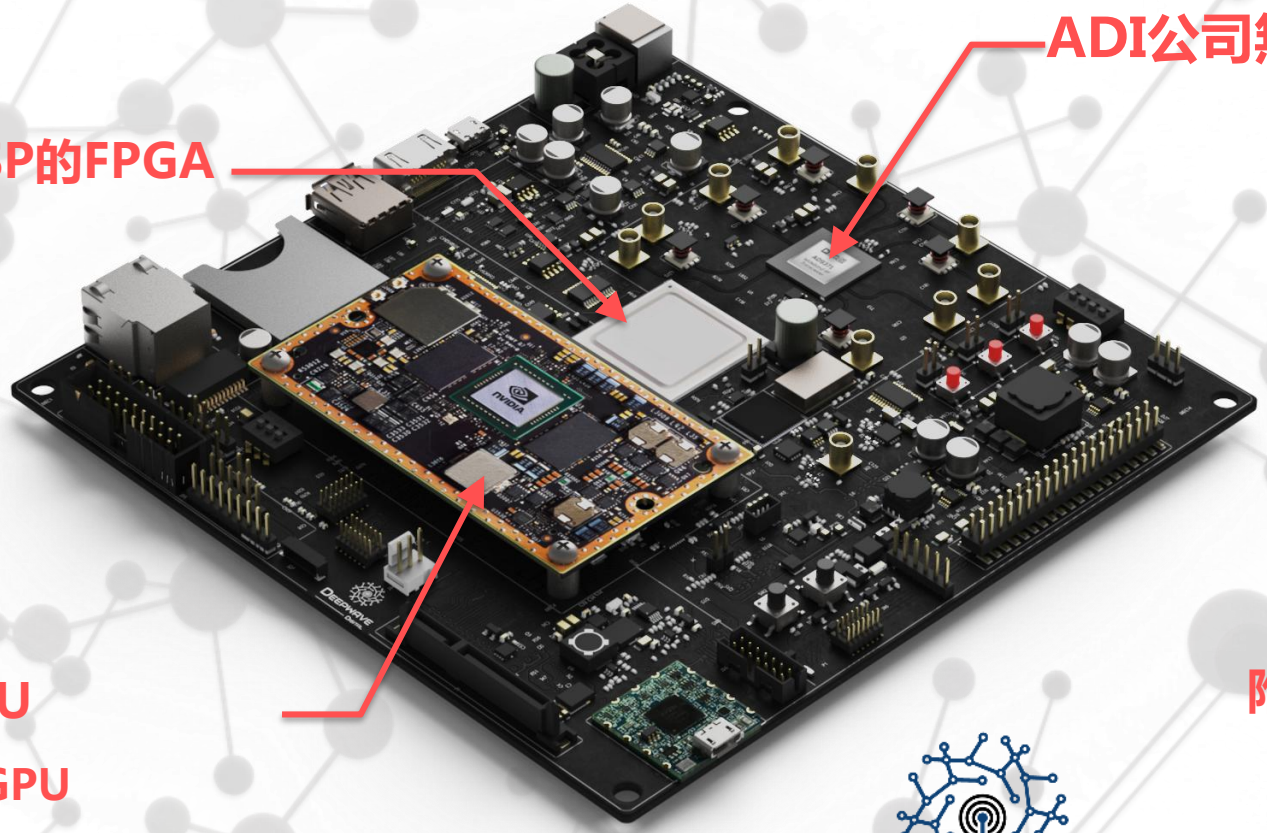
	適應性/升級能力	佈署時間	生命週期成本	即時頻寬	運算/功耗	延遲
CPU						
GPU						
FPGA						
ASIC						

GPU訊號處理能以較低成本和較短的開發時間提供寬頻功能和軟體升級能力

- 必須應對延遲增加的問題（約2微秒）

Deepwave Digital 技術

人工智慧無線電收發器(AIR-T)



支援即時DSP的FPGA

ADI公司無線收發器

- AD9371
- 2x2 MIMO
- 0.3 – 6 GHz
- 內建校準（QEC、LO）

支援AI邊緣運算的GPU

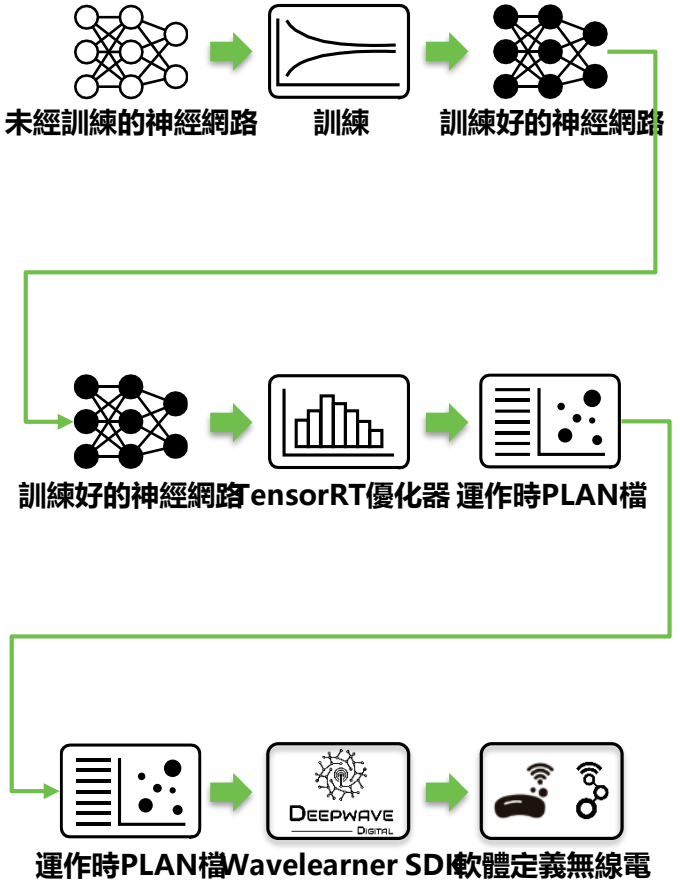
- NVIDIA嵌入式GPU
- 256個GPU核心
- 4個CPU核心(ARM)
- 8 GB記憶體

附加連接

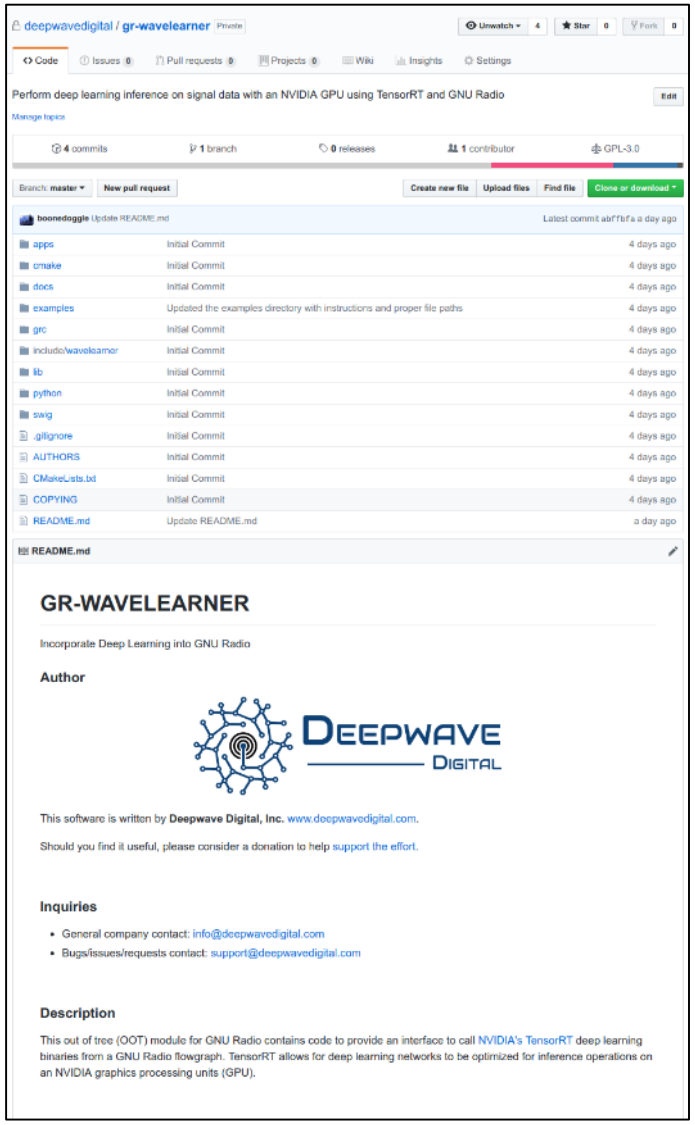
- 1 PPS / 10 MHz用於GPS同步
- 外部LO輸入
- HDMI、USB 2.0/3.0、SATA、乙太網路、SD卡、GPIO

分三步部署神經網路

- **第一步 – 訓練**
  - 使用TensorFlow、MATLAB、Keras、PyTorch等
- **第二步 – 優化**
  - 使用NVIDIA的 TensorRT
- **第三步 – 佈署**
  - 使用Deepwave的GR-Wavelearner和GNU Radio



開源GR-Wavelearner軟體

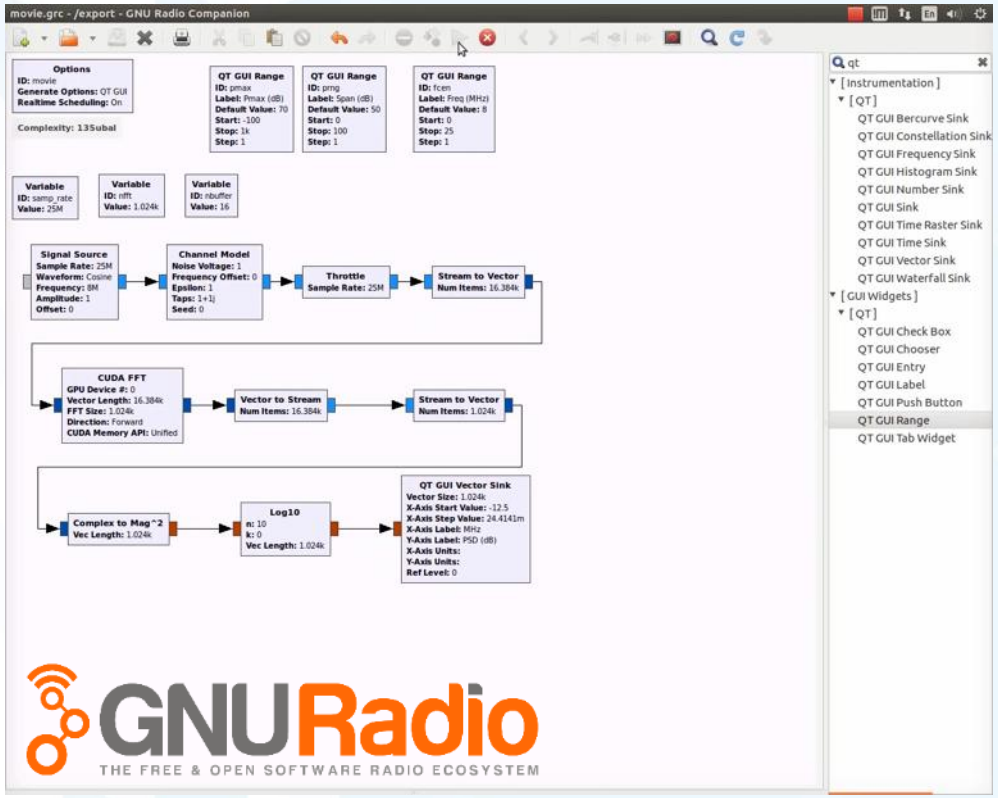


- **目標是說明開源社群在訊號處理應用中輕鬆部署深度學習**
- **完整清楚的讀我檔案以及依賴安裝說明可幫助用戶快速入門**
  - 推薦Ubuntu 16.04，支持Windows 10
  - NVIDA Docker容器18.08*
- **提供的訊號分類器示例：**
  - GNU Radio流程圖
  - Python原始程式碼
  - 在AIR-T和Maxwell上可執行的PLAN檔
  - 用於測試的訊號資料檔案示例
- **支援TensorRT 5.0**
- **可在以下網址獲得：deepwavedigital.com/wavelearner**

* https://docs.nvidia.com/deeplearning/sdk/tensorrt-container-release-notes/rel_18.08.html

GNU Radio–軟體定義無線電(SDR)框架

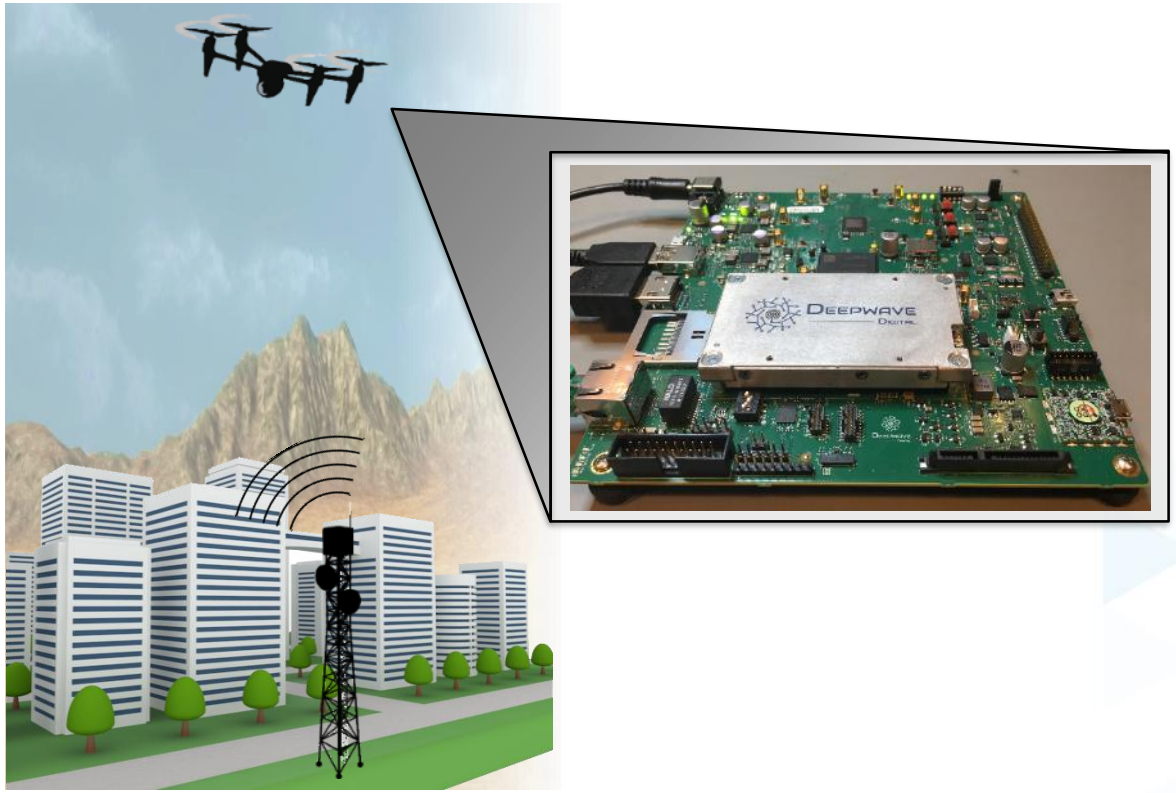
- **受歡迎的開源軟體定義無線電(SDR)工具包：**
  - 射頻硬體可選
  - 可以運作完整軟體模擬
- **Python API**
  - 底層是C++
- **輕松建立DSP演算法**
  - 自定義用戶塊
- **主要使用CPU**
  - 高級並行指令
  - 最新發展：RFNoC支援FPGA處理
- **Deepwave正在整合GPU對DSP和ML的支援**



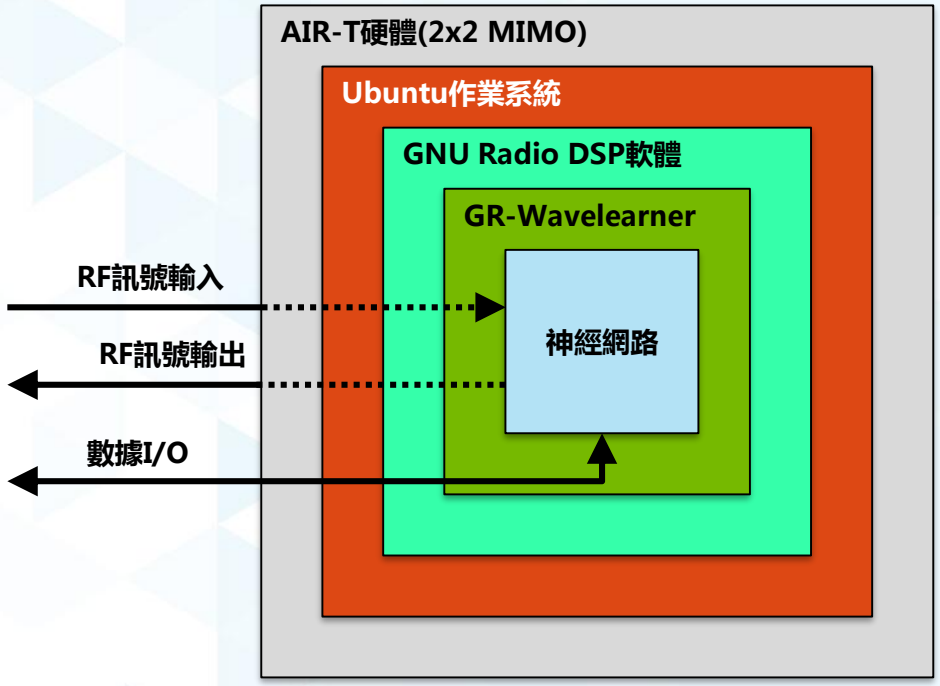
Deepwave解決方案和平台

在射頻應用中無縫部署深度學習

人工智慧無線電收發器(AIR-T)



GR-Wavelearner軟體

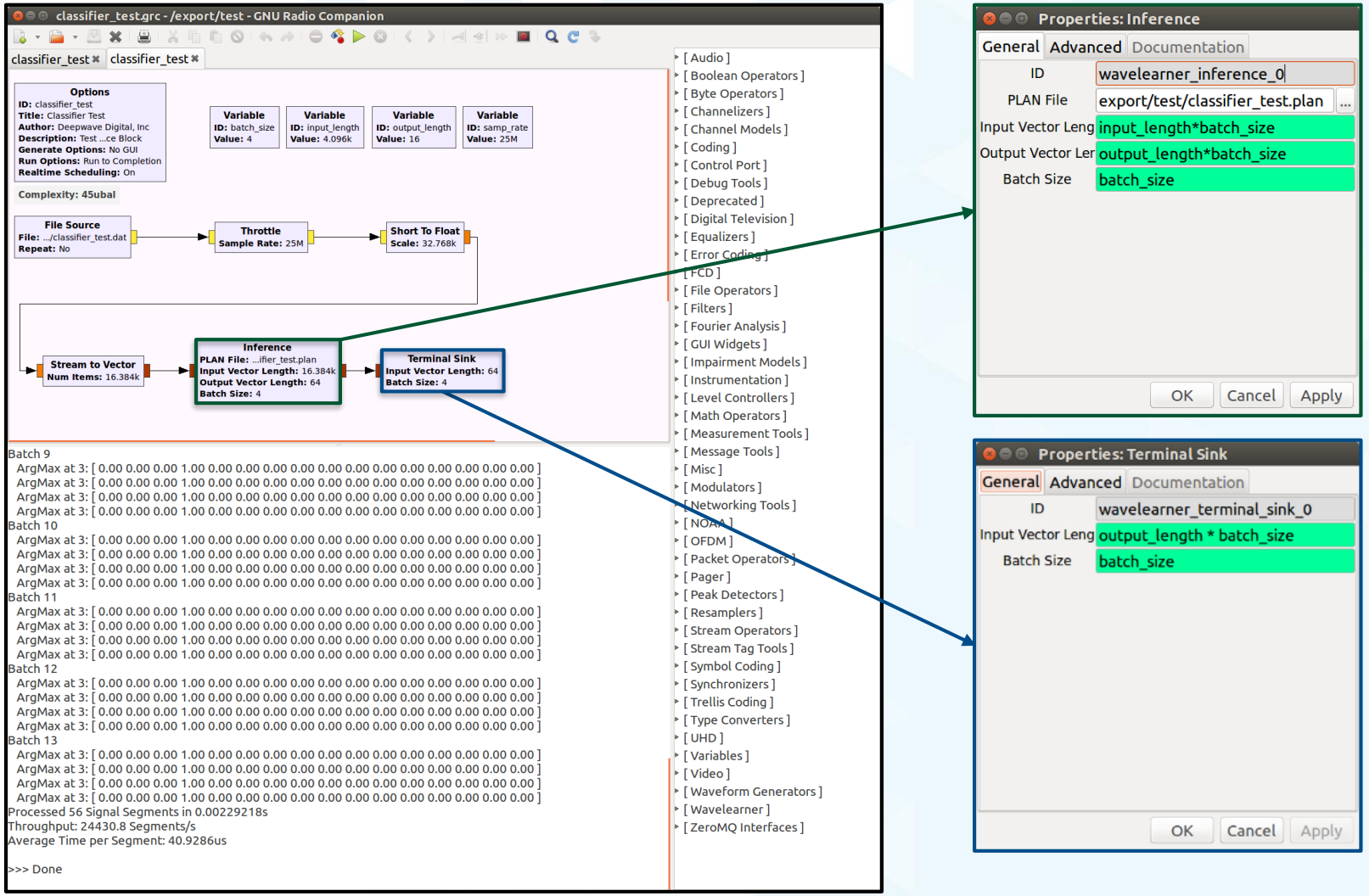


AIR-T預裝有部署所需的全部軟體工具

18

GR-Wavelearner

- 用於GNU Radio的樹外(OOT)模組
- 支援使用者將深度學習輕鬆整合到訊號處理中
- C++和Python API
- 開源GPLv3許可證
- 當前有兩個塊：
  - 推理 – 用於GNU Radio的TensorRT包裝器
  - 終端接收器 – 用於顯示分類器輸出的Python模組



訓練到佈署流程

- 工作流程利用TensorRT進行部署
- 支持在各種框架上進行訓練

原生支援TensorRT

TensorFlow
MathWorks
K

通過ONNX支援**

Caffe2
Cognitive Toolkit
PaddlePaddle
PyTorch
Chainer
mxnet

*深度神經網路

**開放式神經網路交換

Deepwave的訓練到佈署流程

TensorFlow示例

第一步：凍結圖形（使變數成為常量）

第二步：將DNN*模型轉換為UFF檔

第三步：將UFF檔轉換為PLAN檔

注意：此步必須在部署架構上完成

注意事項

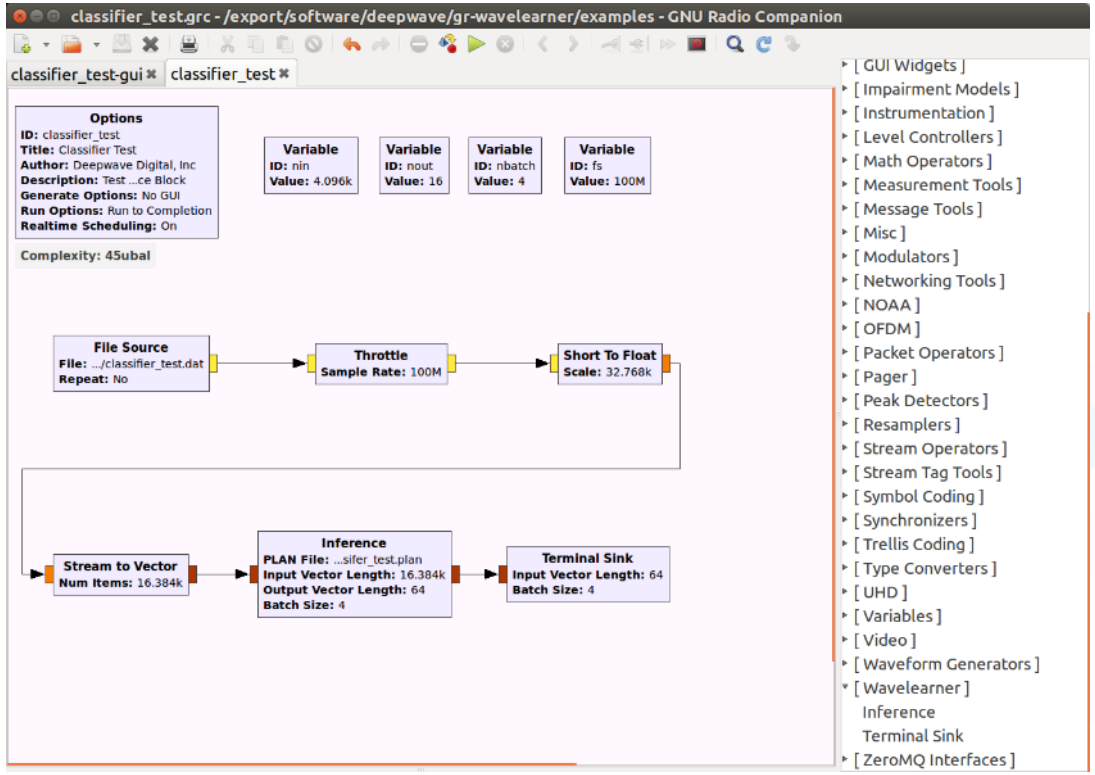
- 並非所有層都受支援，但支援最常見的層
- PLAN檔必須在部署架構上建立
  - Python轉換在ARM (Jetson)上不可用
  - PLAN文件的傳輸能力有限

*深度神經網路

Deepwave的訓練到佈署流程

訓練 → 優化 → 部署

GNU Radio配套GUI



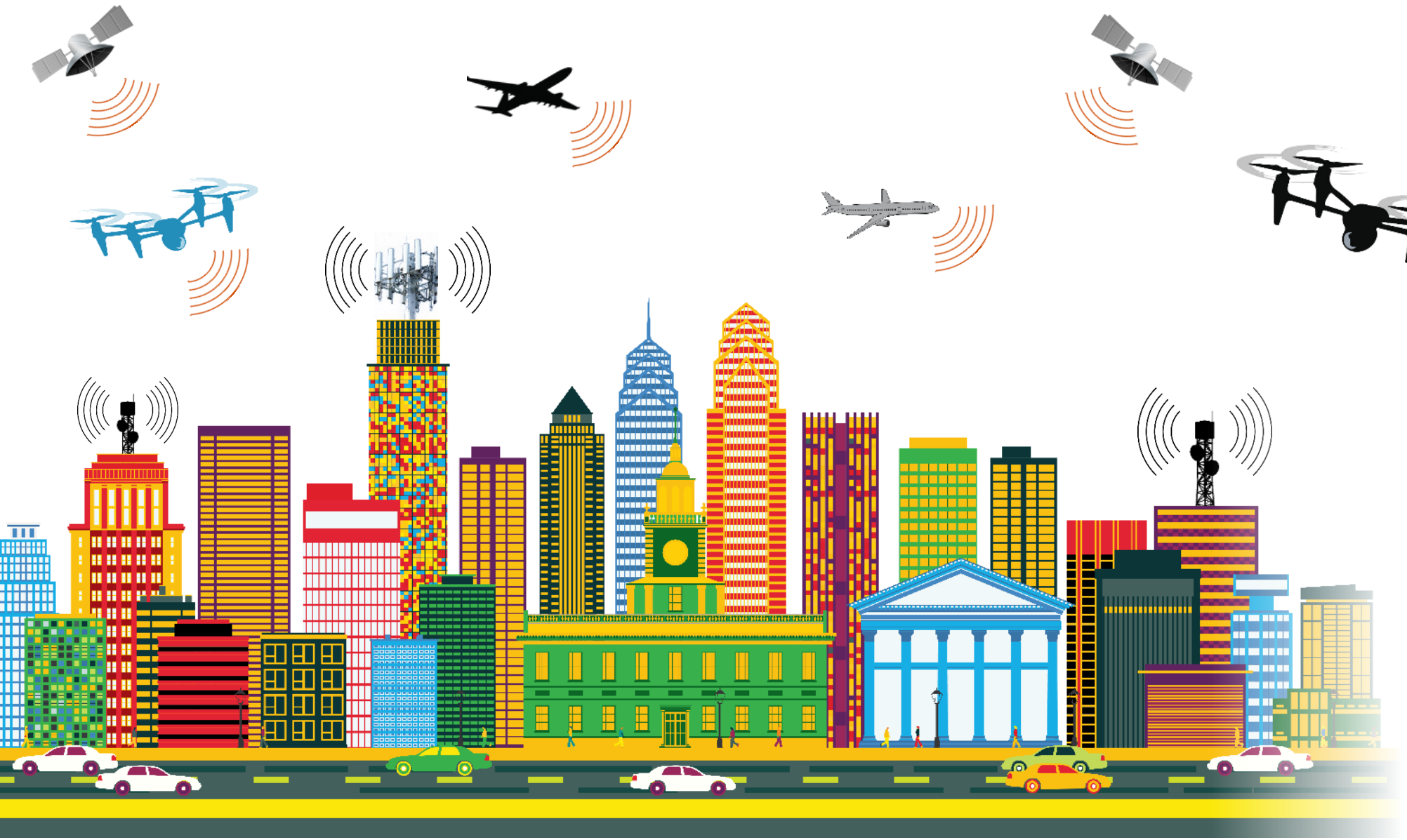
Python API
35行代碼實現即時DNN* RF系統！

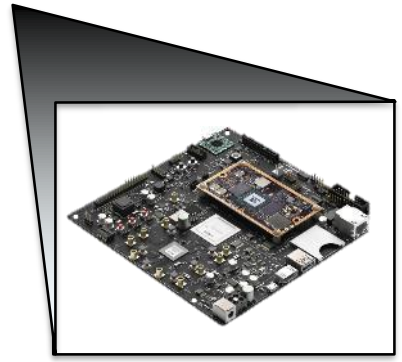
```python
#!/usr/bin/env python2
from gnuradio import blocks
from gnuradio import gr
import wavelearner

class ClassifierTest(gr.top_block):
    def __init__(self):
        gr.top_block.__init__(self)

        # Define signal processing functions
        self.source       = blocks.file_source(gr.sizeof_short, inputdata, False)
        self.throttle     = blocks.throttle(gr.sizeof_short, fs, True)
        self.type_convert = blocks.short_to_float(1, norm)
        self.buffer       = blocks.stream_to_vector(gr.sizeof_float, nbatch*nin)
        self.inference    = wavelearner.inference(planfile, nin*nbatch, nout*nbatch, nbatch)
        self.terminal_out = wavelearner.terminal_sink(nout * nbatch, nbatch)

        # Connect functions in flowgraph
        self.connect((self.source, 0), (self.throttle, 0))
        self.connect((self.throttle, 0), (self.type_convert, 0))
        self.connect((self.type_convert, 0), (self.buffer, 0))
        self.connect((self.buffer, 0), (self.inference, 0))
        self.connect((self.inference, 0), (self.terminal_out, 0))

# Define Settings
inputdata = '/path/to/classifier_test.dat'
planfile  = '/path/to/classifier_test.plan'
fs        = 100e6
nin       = 4096
nout      = 16
nbatch    = 4
norm      = 32768
tb        = ClassifierTest()
tb.start()
tb.wait()
```


*深度神經網路

訊號檢測和分類示例

多發射器環境情況





接收器利用深度學習監視擁塞頻譜

雷達訊號探測器模型：發射訊號

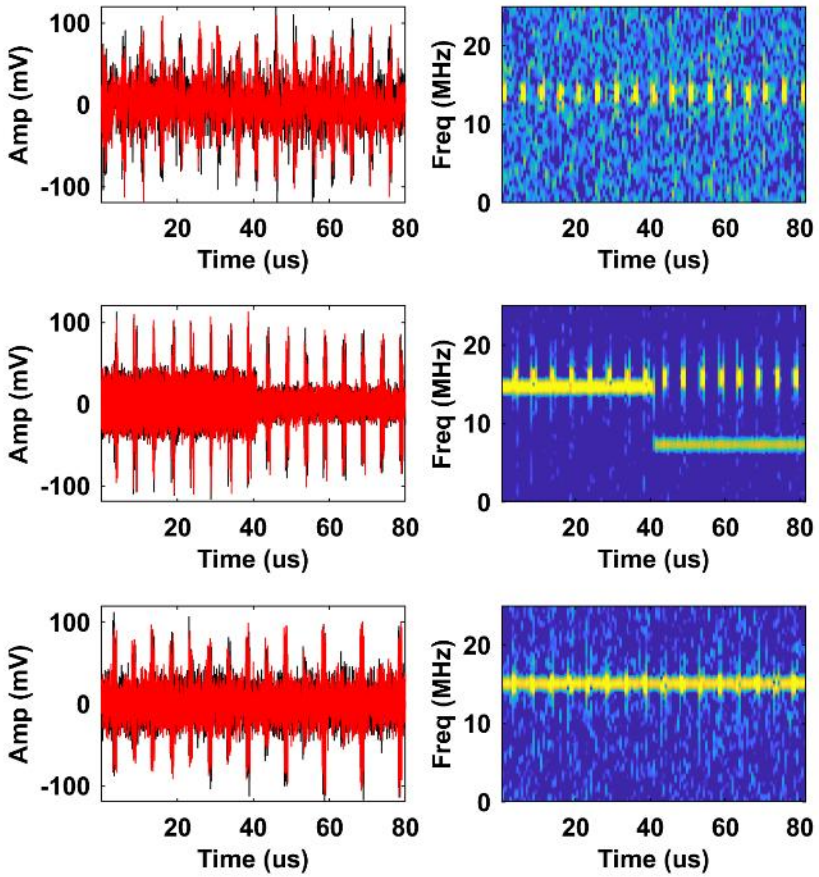
雷達波形	無	干擾	監控	地面(LFM1)	地面(LFM2)	MTI	空中（中PRF）	空中（高PRF）	地面（弗蘭克碼）	海上（短距離）	海上（長距離）	海上（長距離）	地面(NLFM1)	地面(NLFM2)	地面(NLFM3)
線性脈衝			X	X	X					X	X	X			
非線性脈衝													X	X	X
相位編碼脈衝									X						
脈衝多普勒						X	X	X							

利用標稱雷達訊號顯示的技術演示

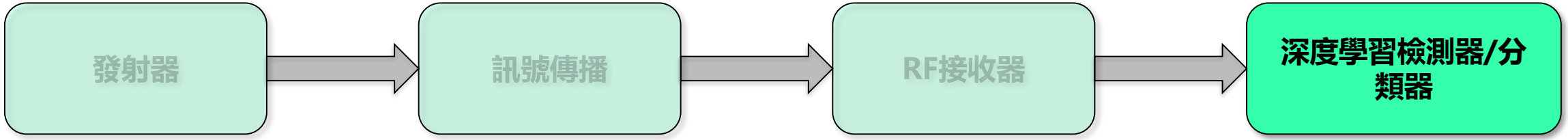
- 適用於通訊、蜂巢和其他RF協議的方法

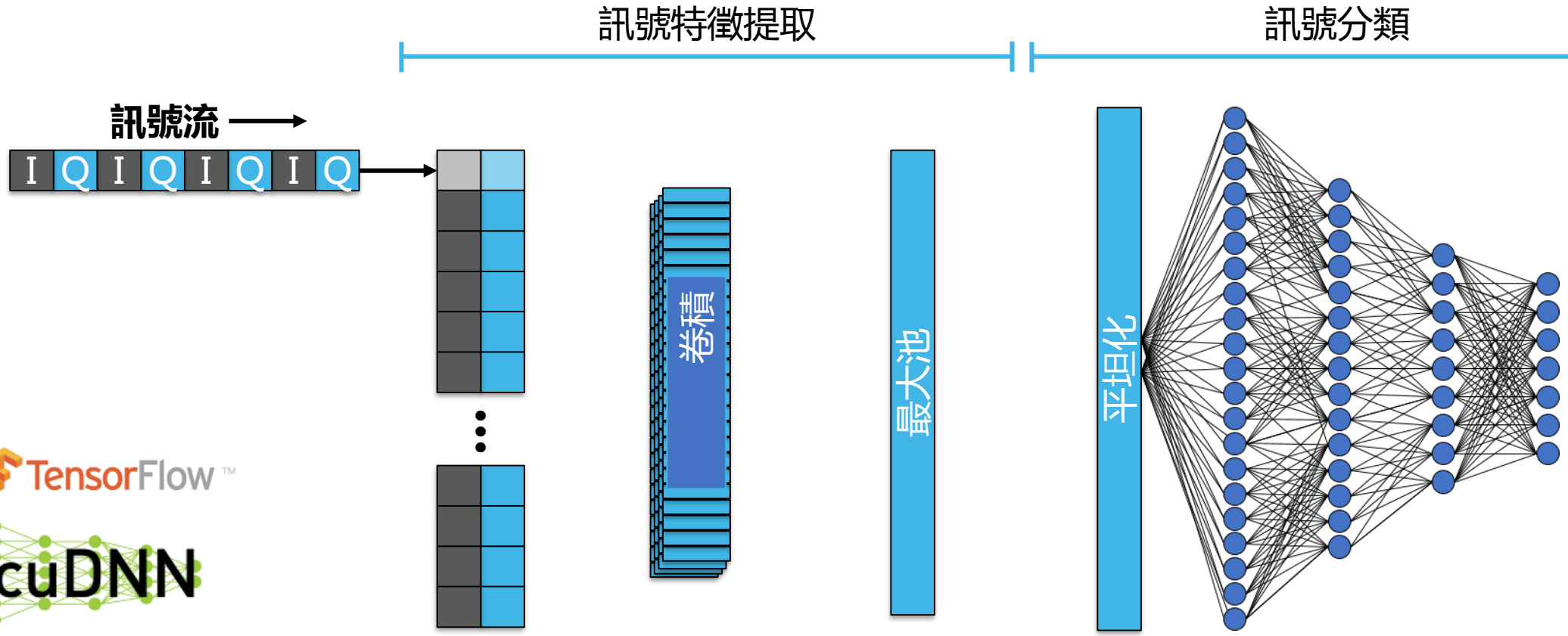
資料集概述

- 目標：開發深度學習分類器，檢測低於雜訊底層的訊號
  - 需要對有干擾和無干擾的雜訊資料進行訓練
- SNIR從-35 dB掃描到20 dB（增量為1 dB）
  - 每個SNIR有1000個訓練段
  - 每個SNIR有500個推理段
  - 每段最多3個干擾源



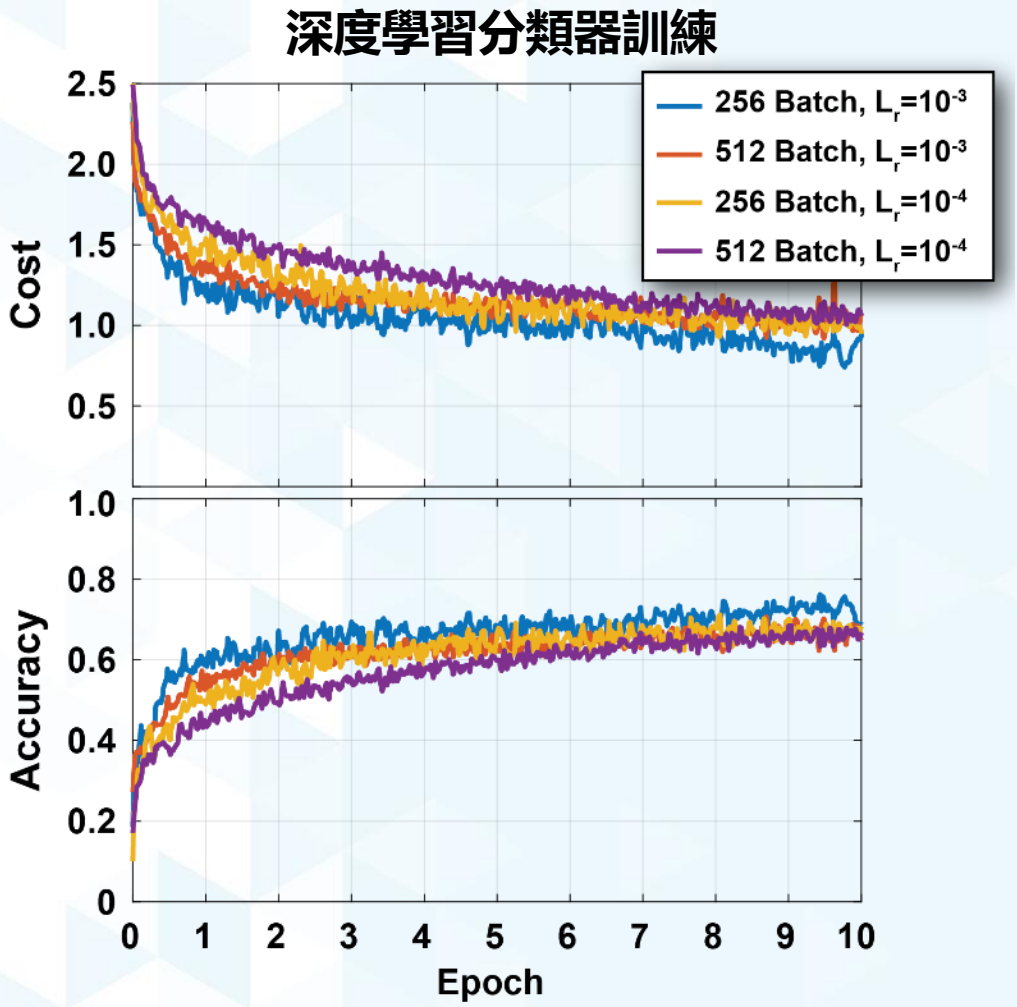
雷達訊號探測器模型：分類器示例



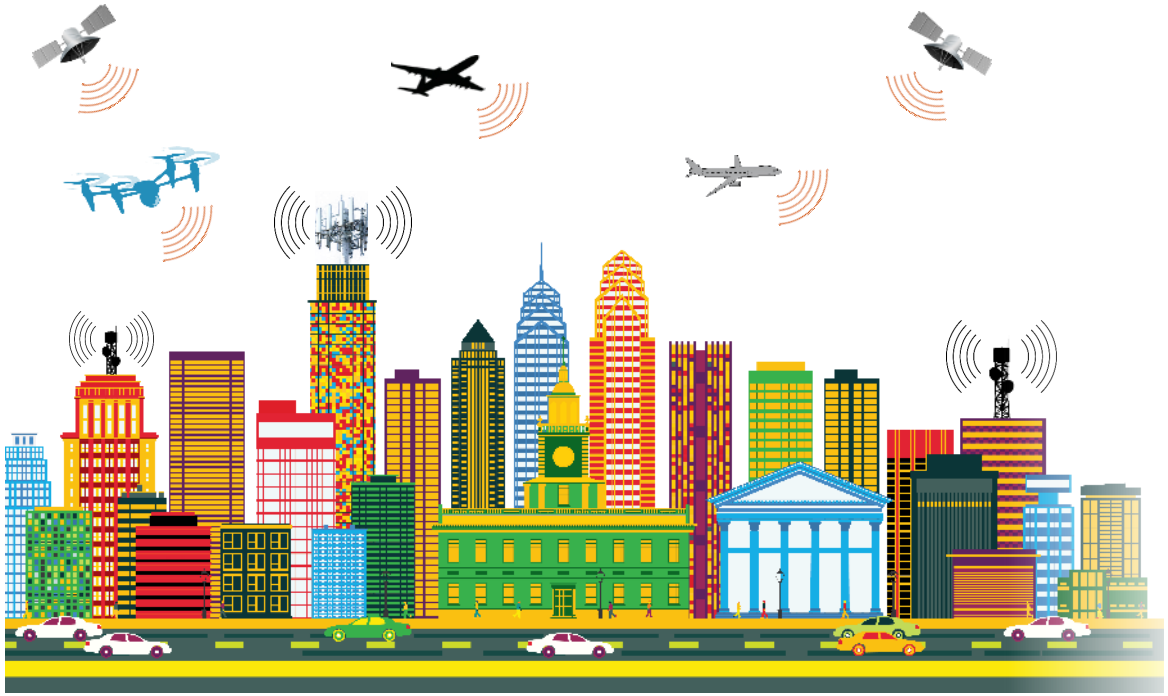


訓練過程和進度

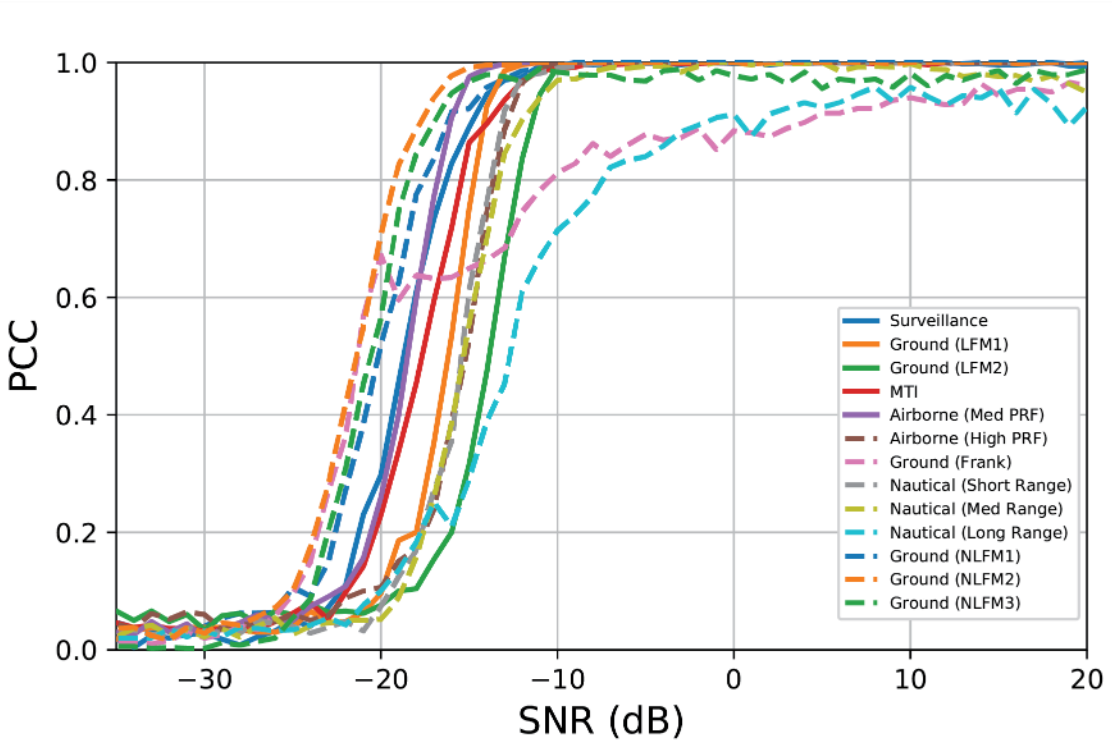
- 每個SNR有1000個訓練段
  - 55個不同SNR值
- 低SNR值訓練可提高檢測靈敏度
- 由於在超低SNR值下進行訓練，無法獲得100%的準確性
- Softmax交叉熵
- Adam優化器



接收器工作特性(ROC)曲線

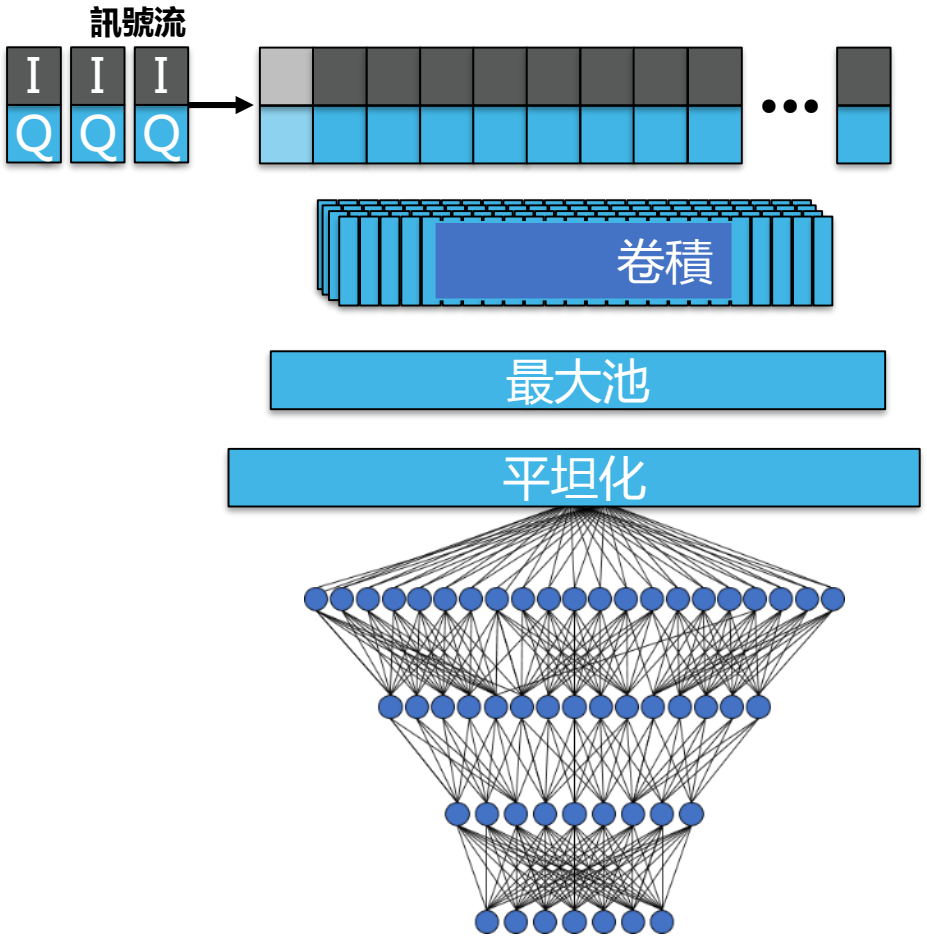


所有訊號正確分類的概率



在嵌入式GPU上對深度學習推理進行基準測試

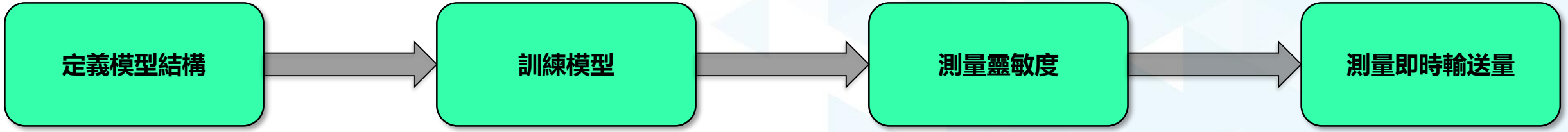
關鍵性能參數



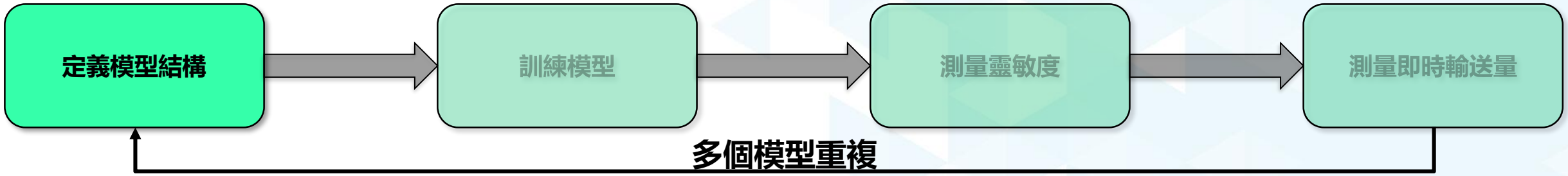
- 如何判斷DNN模型的好壞？
  - **高靈敏度** – 檢測低功率訊號
  - **低誤報率** – 使誤報率最小
  - **高即時頻寬**
  - **運算需求低**
  - **低延遲**

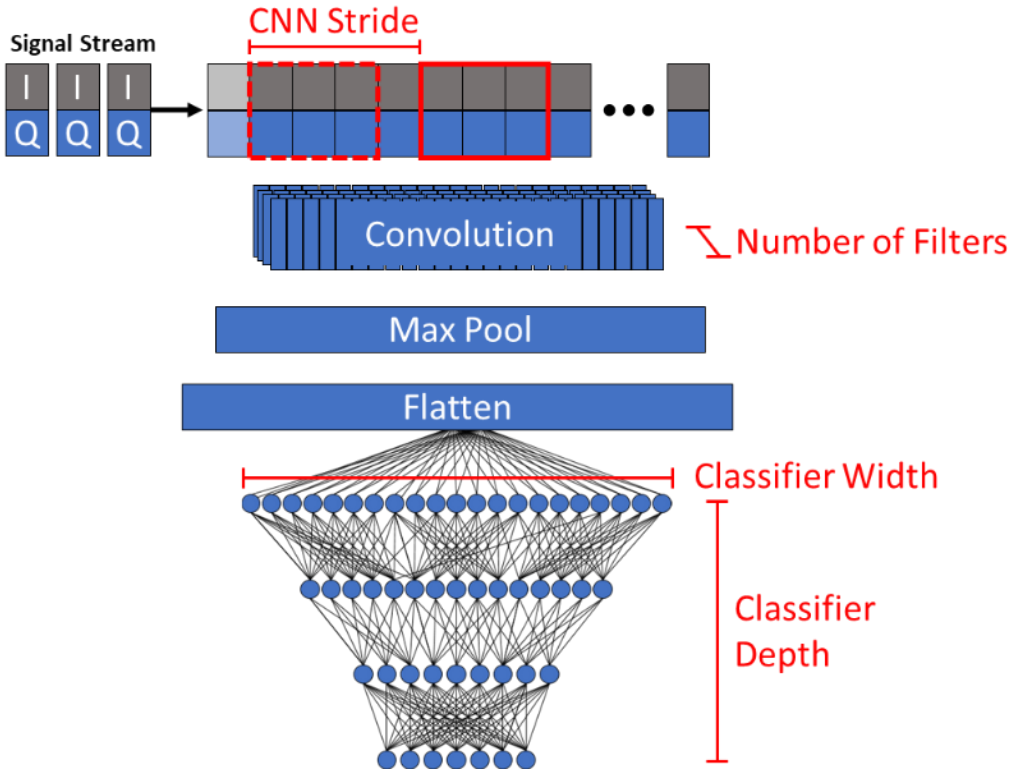
- 這些關鍵性能參數大多數是對抗性的

性能基準測試設置



性能基準測試設置

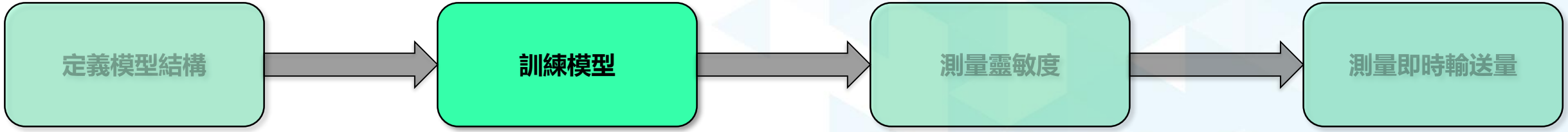




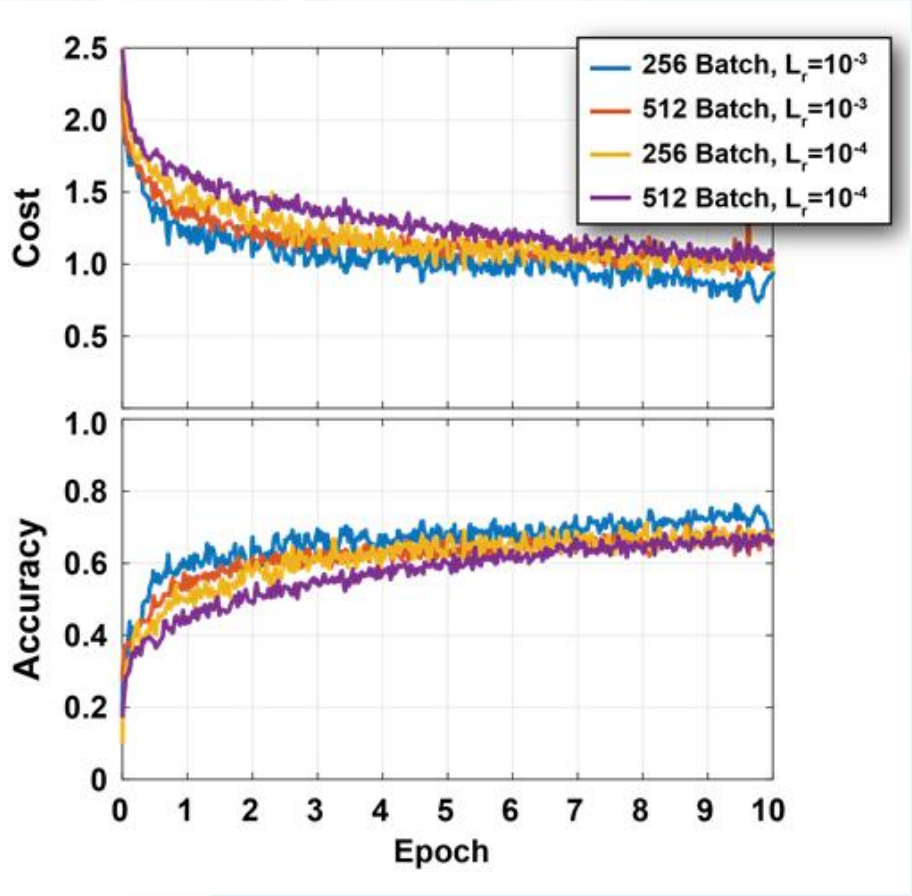
模型調整變數

	最小值	最大值	總計
CNN步長	1	16	9
篩檢程式數量	4	256	7
分類器第1層寬度	64	128	3
分類器第2層寬度	32	64	3
分類器第3層寬度	0	64	2
批次大小	1	256	8
		測試模型總計	728

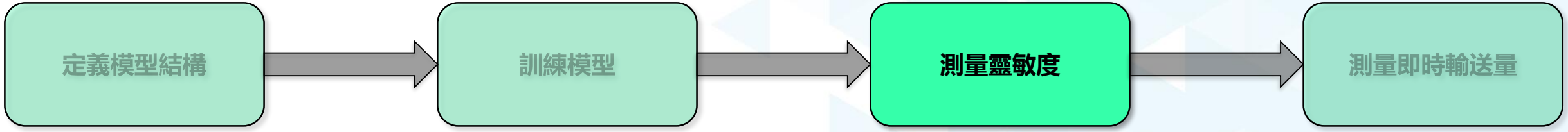
性能基準測試設置



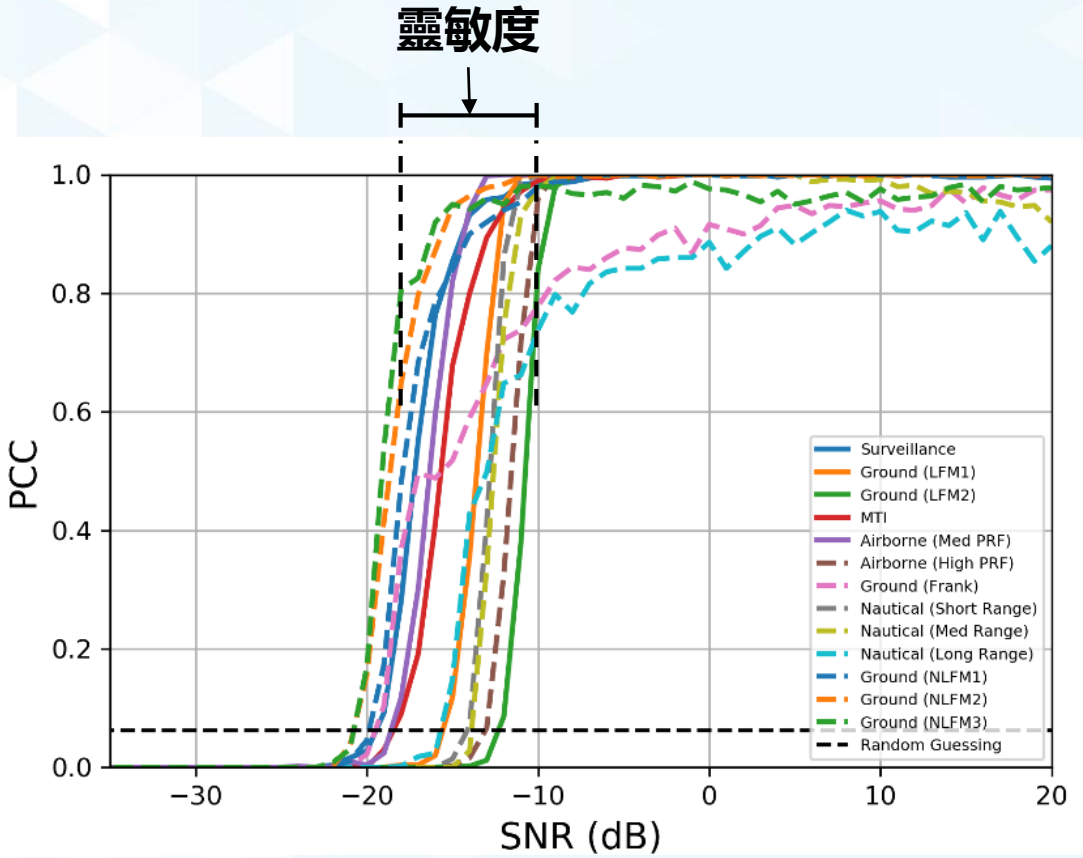
- 每個SNR有1000個訓練段
  - 55個不同SNR值
- Softmax交叉熵
- Adam優化器
- Quadro GP100 GPU
- 為每個模型建立UFF檔



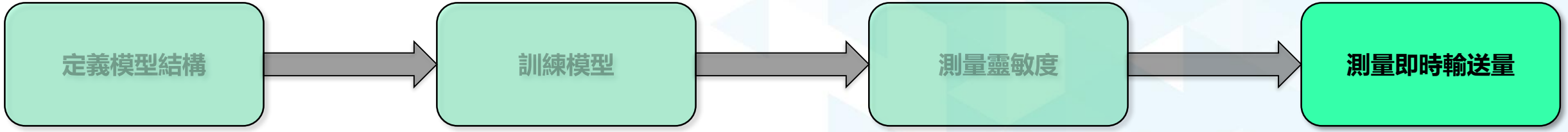
性能基準測試設置



- 運算每個模型的接收器工作特性(ROC)曲線
- 將靈敏度定義為所有訊號類型的中值PCC = 50%



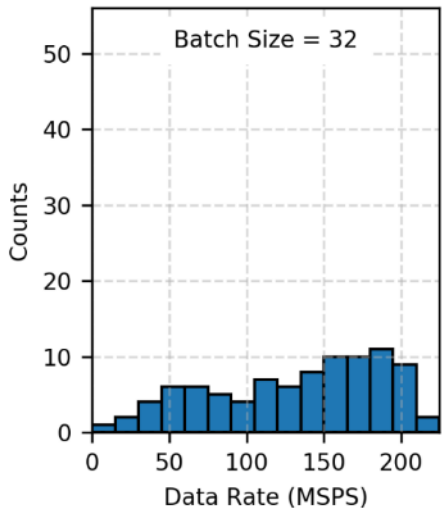
性能基準測試設置



- 為每個測試平台建立TensorRT PLAN檔
- 將訊號數據載入到RAM
- 將未經調節的資料流程傳輸到GR-Wavelearner

- 在兩個位置探測資料速率：
  1. 整個過程的總數據速率
     - 處理的位元組數/牆上時間
  2. work() 函數中的運算資料速率
     - 處理的位元組數/運算時間

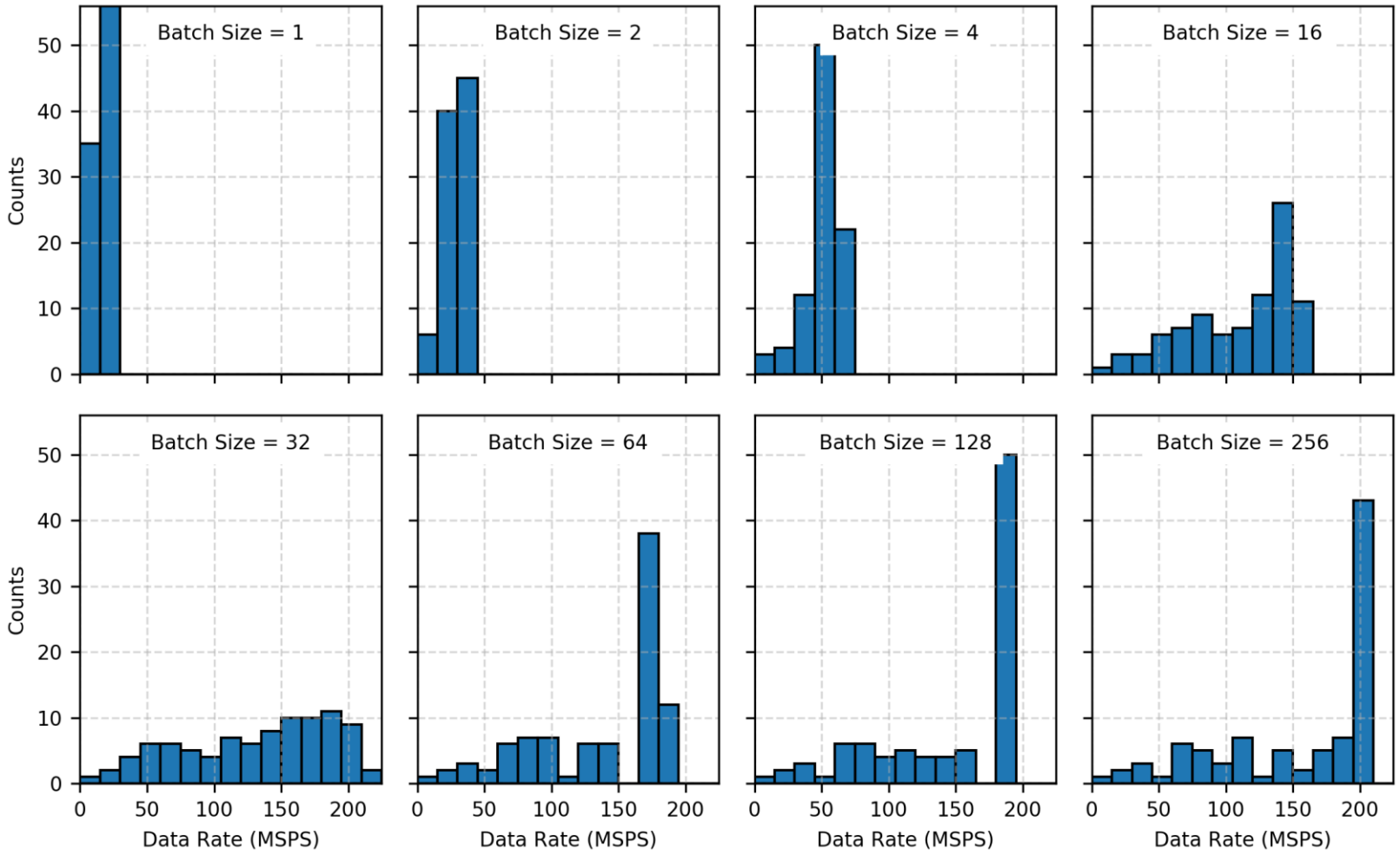
AIR-T (Tegra TX2)的資料速率基準



- 測試了91種不同的CNN分類器模型
- 8個不同批次大小的最大即時推理資料速率
- 利用AIR-T能夠達到200 MSPS（實際）

AIR-T

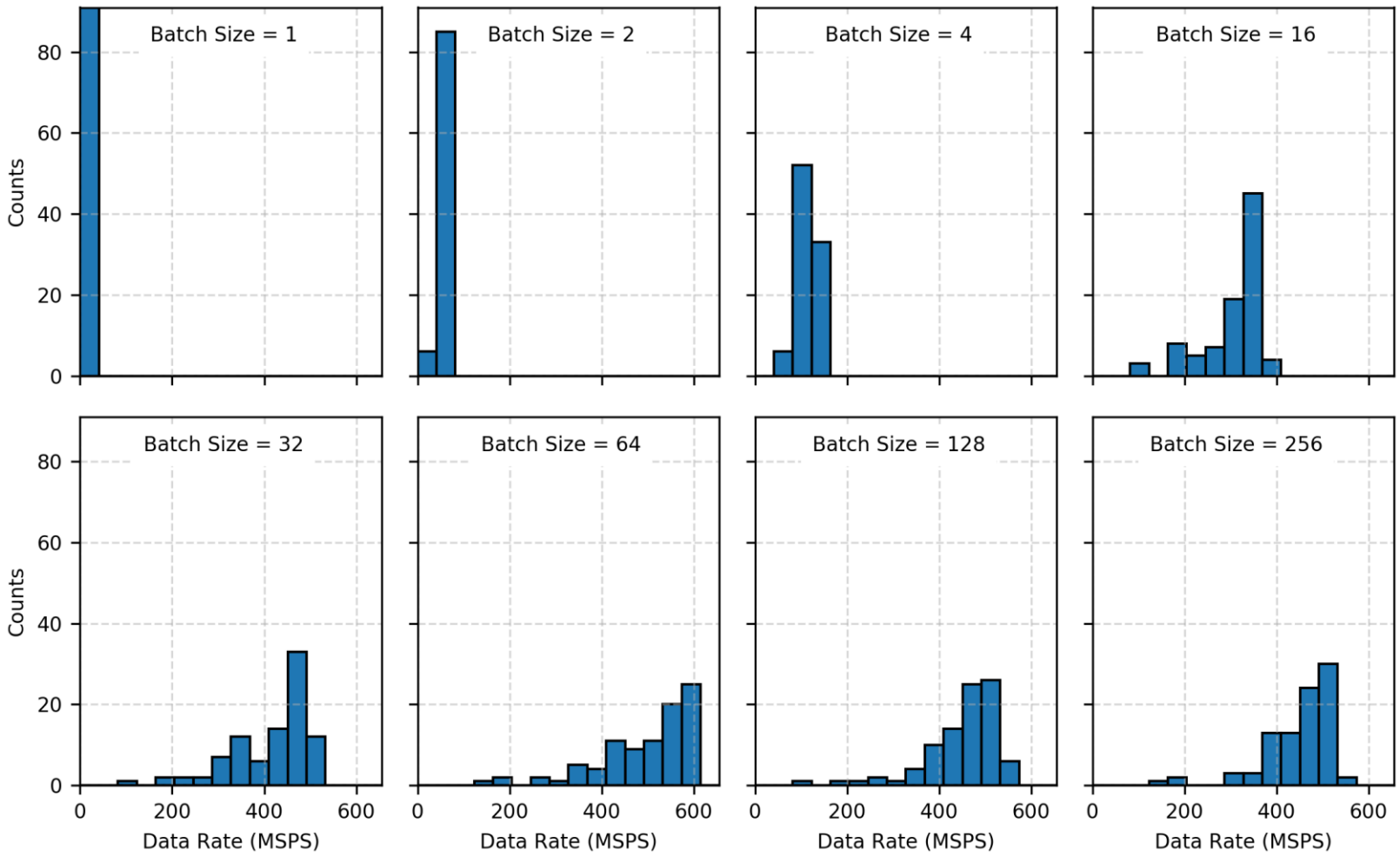
AIR-T (Tegra TX2)的資料速率基準



- 測試了91種不同的CNN分類器模型
- 8個不同批次大小的最大即時推理資料速率
- 利用AIR-T能夠達到200 MSPS（實際樣本）

AIR-T

桌上型電腦(Quadro P100)的資料速率基準



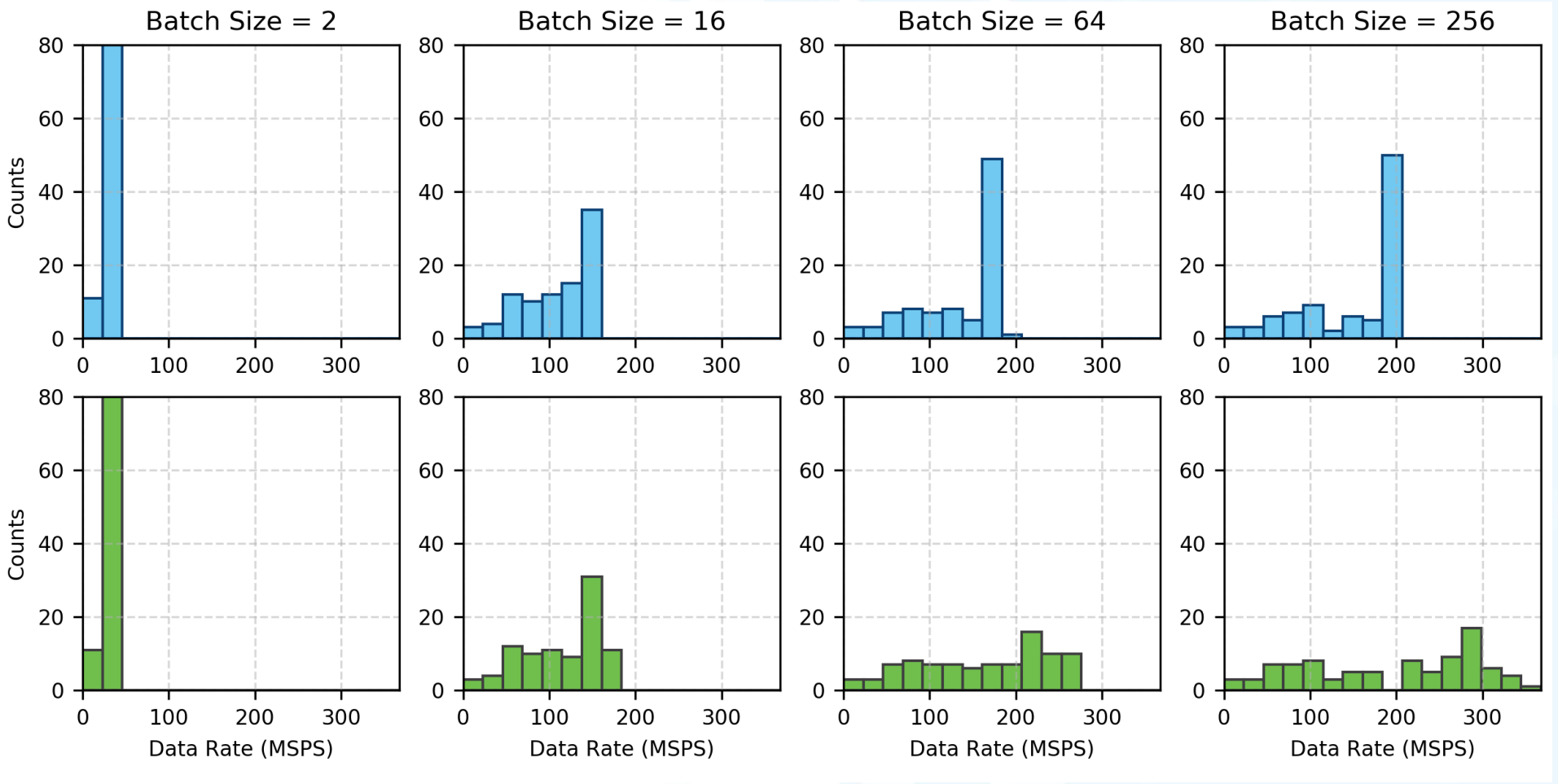
- 測試了91種不同的CNN分類器模型
- 8個不同批次大小的最大即時推理資料速率
- 使用統一記憶體可提高輸送量

桌上型電腦(GP100)

AIR-T的牆上時間與運算時間

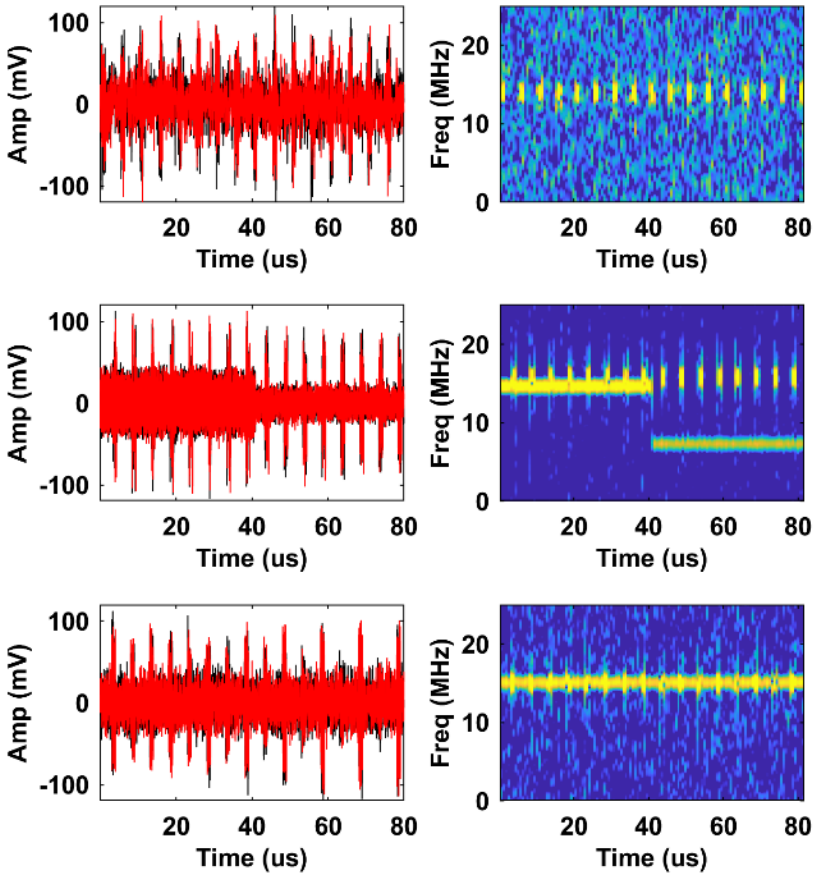
牆上時間

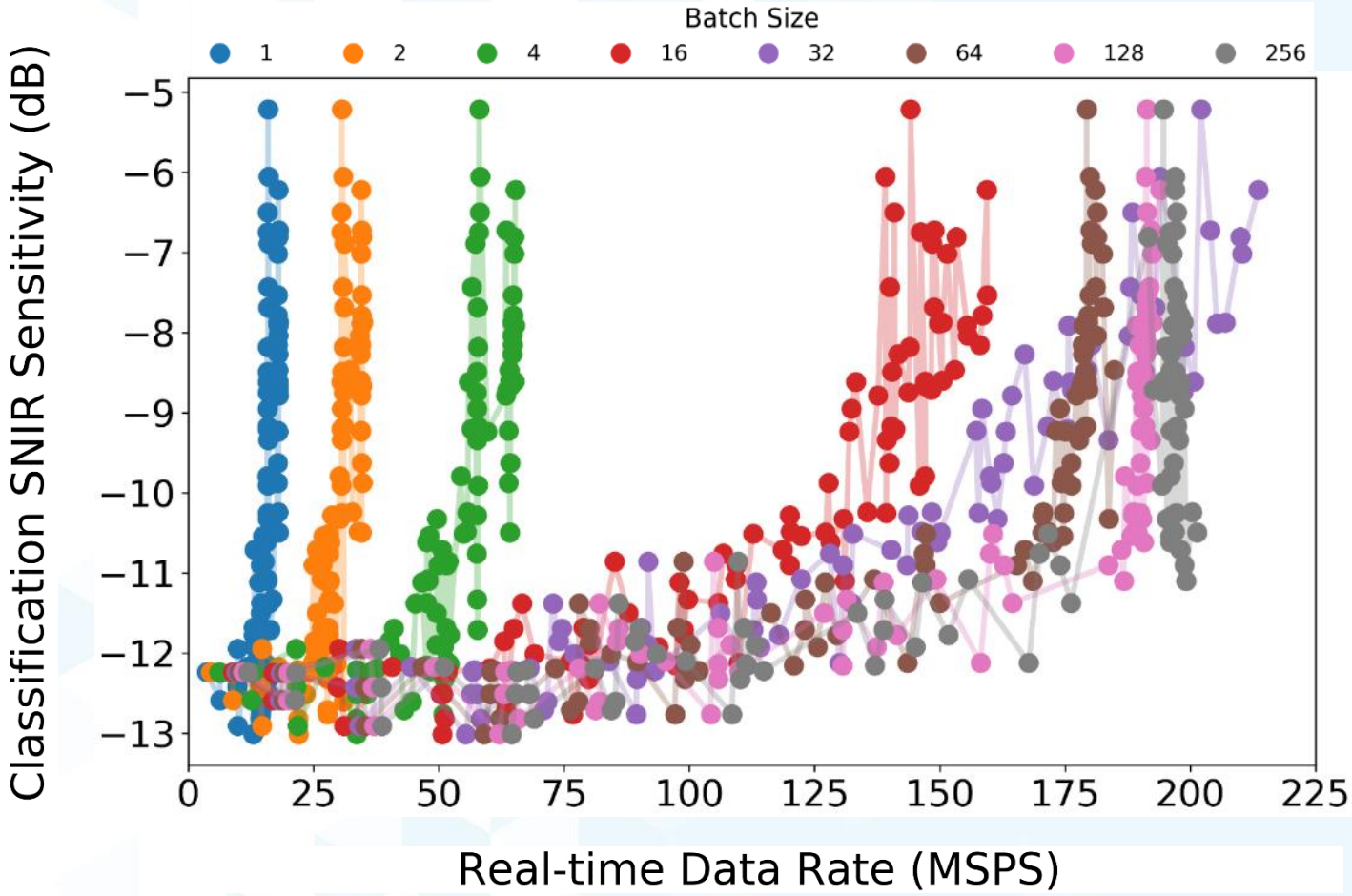
運算時間



即時資料速率受GNU Radio開銷限制

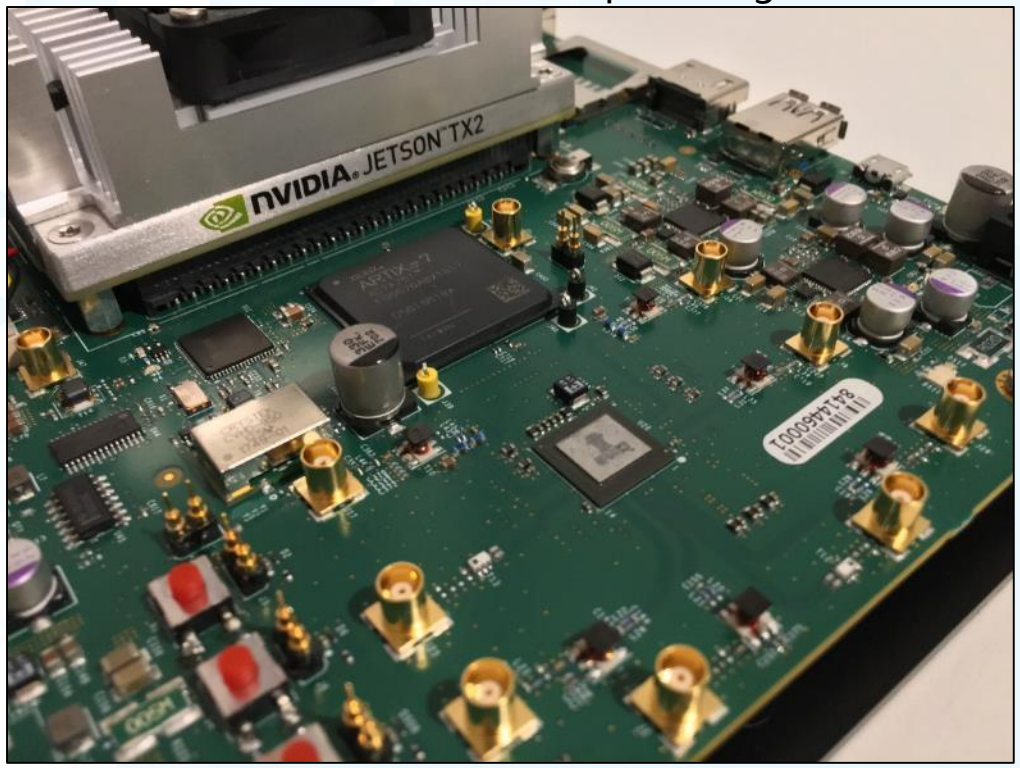
模型精度基準


Amp (mV)
Freq (MHz)
Time (us)


Batch Size
1
2
4
16
32
64
128
256
Classification SNIR Sensitivity (dB)
Real-time Data Rate (MSPS)

總結

欲瞭解更多資訊，請瀏覽www.deepwavedigital.com/sdr



- 訊號處理中的深度學習正在興起
  - 算法可應用於訊號的資料內容或訊號本身
- 高頻寬需求推動邊緣解決方案
- Deepwave開發了AIR-T
  - 帶MIMO收發器的邊緣運算推理引擎
  - FPGA、CPU、GPU用於運算
- GR-Wavelearner軟體：
  - 適用於軟體無線電的開源推理引擎
  - 現可在Deepwave的GitHub頁面上獲得
- 基準測試分析表明，採用GR-Wavelearner的AIR-T能夠以200 MSPS的即時資料速率進行訊號分類推理
  - 未來版本可能會予以改進

謝謝觀看

- ADI 亞洲地區技術支援熱線： 886-2-2650 2888
- ADI 亞洲地區技術支援信箱：cic.asia@analog.com
- ADI 樣品申請網址：http://www.analog.com/sample