# MCU助您跨界机器视觉与人工智能

宋岩

ROCKY.SONG@NXP.COM

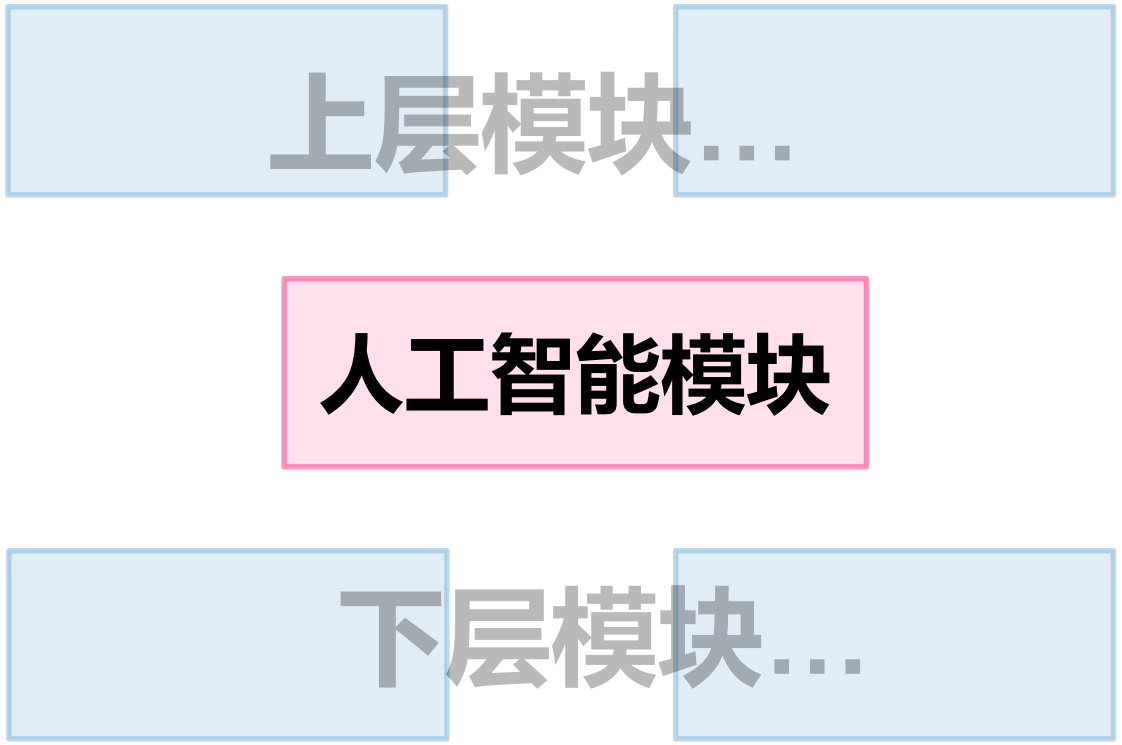
系统&应用工程师

SECURE CONNECTIONS
FOR A SMARTER WORLD

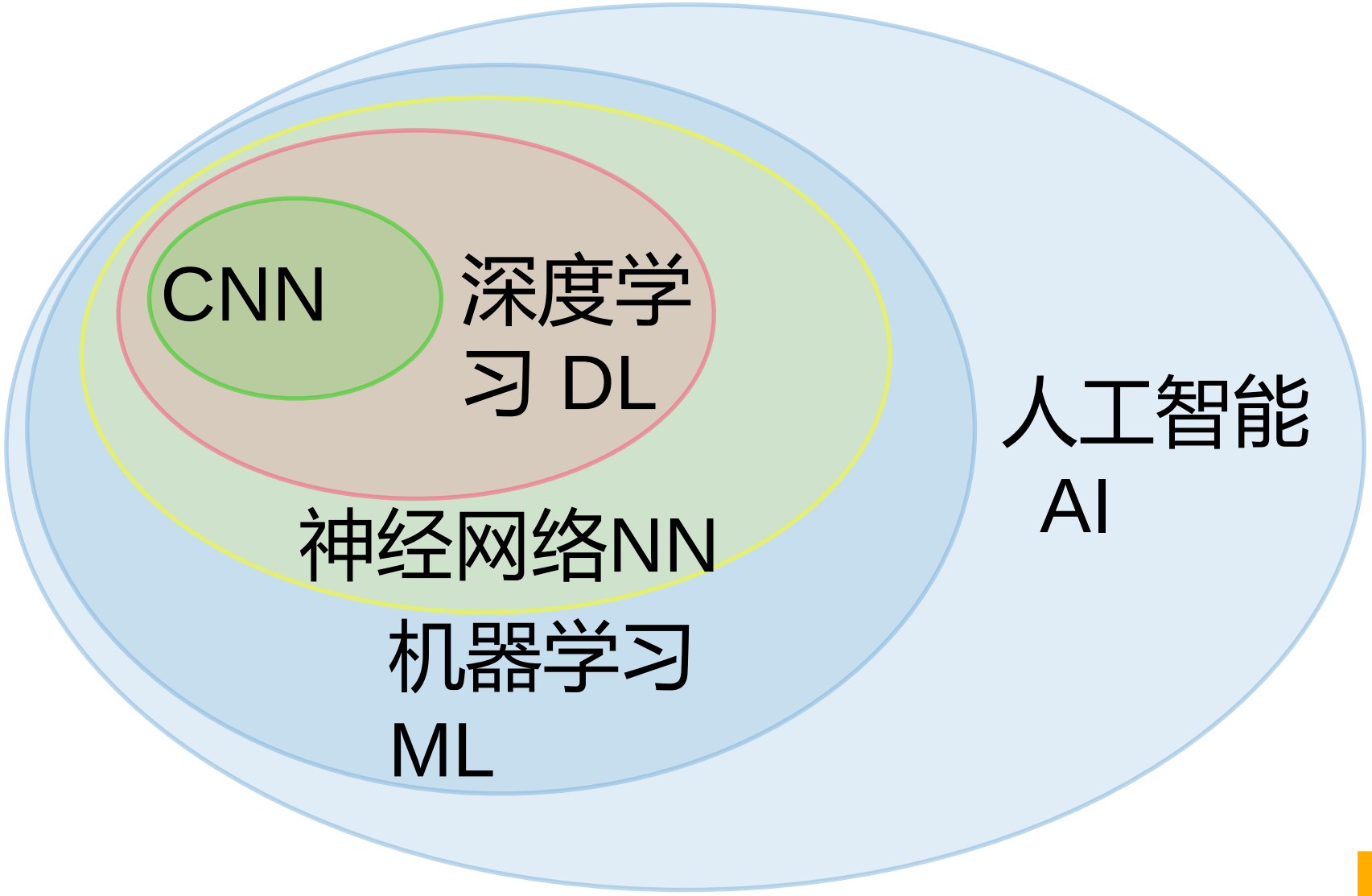
# 在MCU上应用人工智能

# 人工智能与嵌入式系统

- 系统是主体
- 人工智能是属性
- 以一个模块来呈现
- 提供新功能
- 亦可改进现有功能

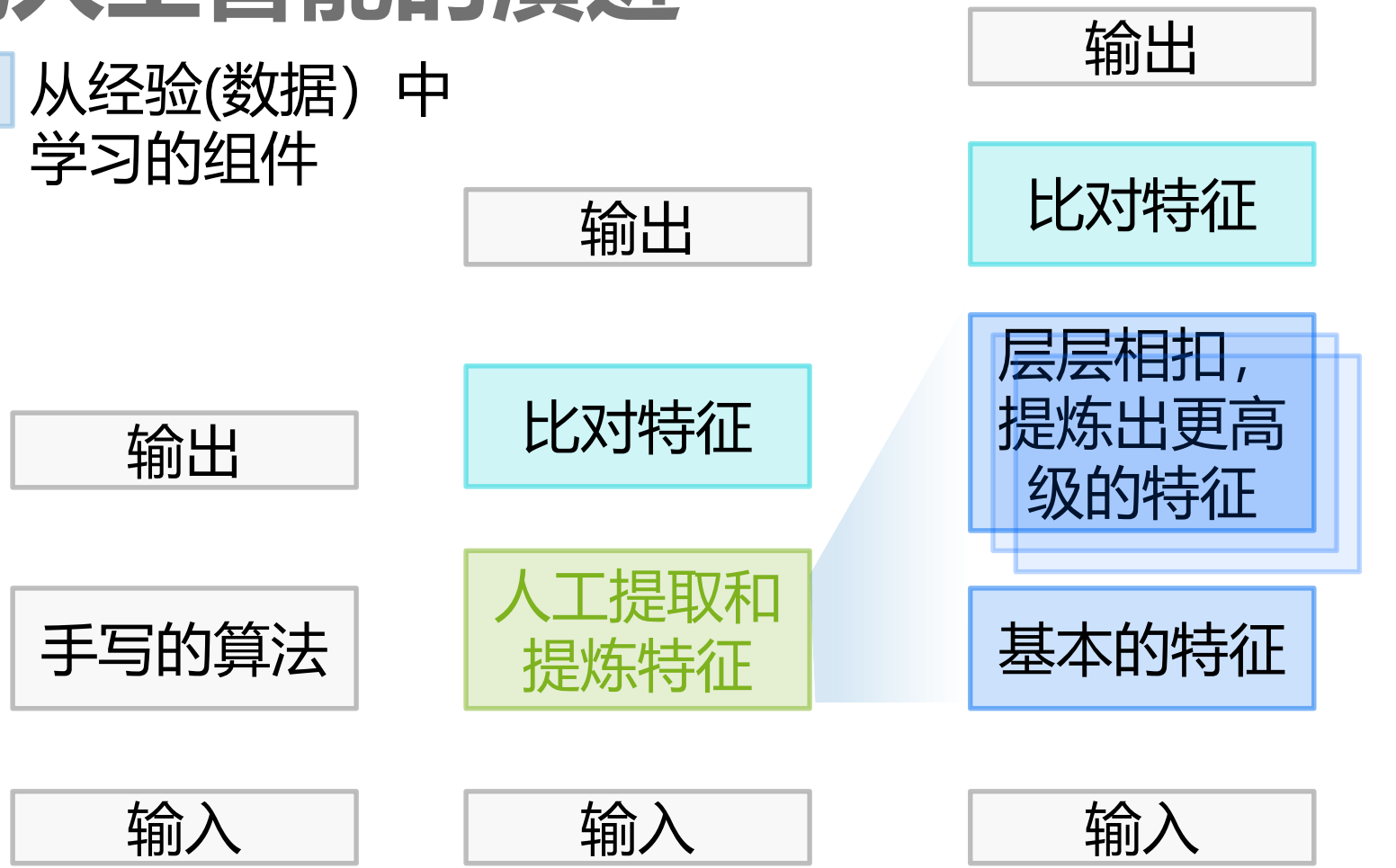


# 人工智能（AI)



# 实现人工智能的演进

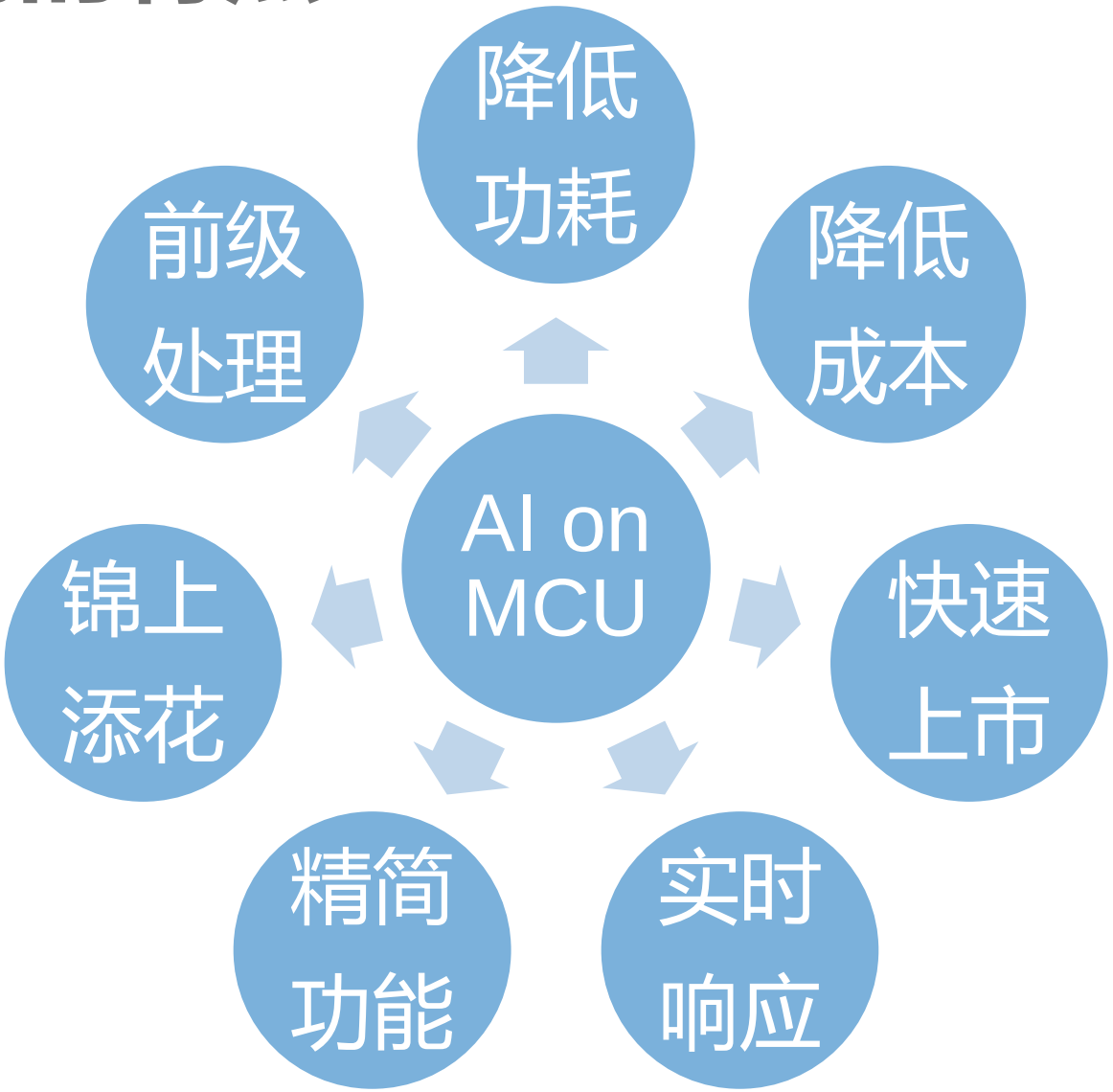
xx 从经验(数据）中学习的组件




DL模型**自动**找出特征，多达成百上千上万!

DL模型基于多层**神经网络**

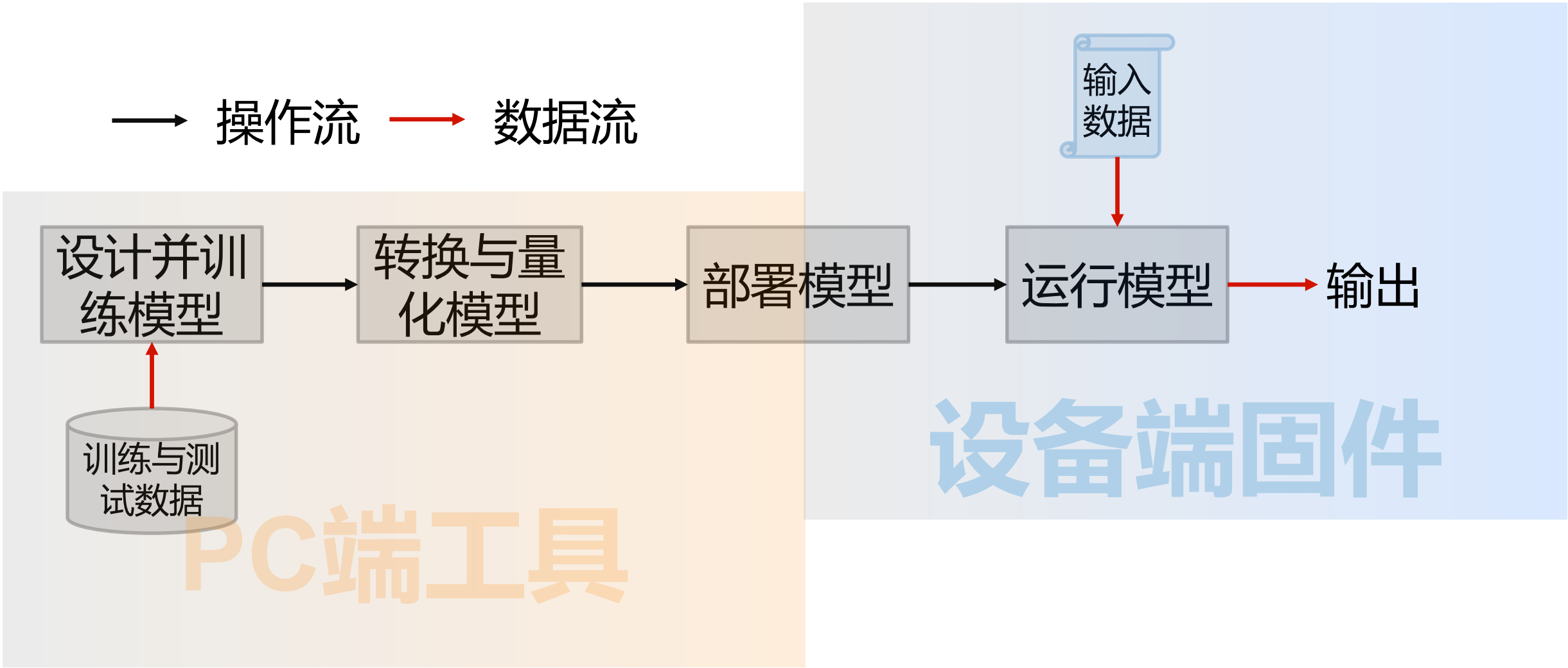
# MCU上AI应用的特点



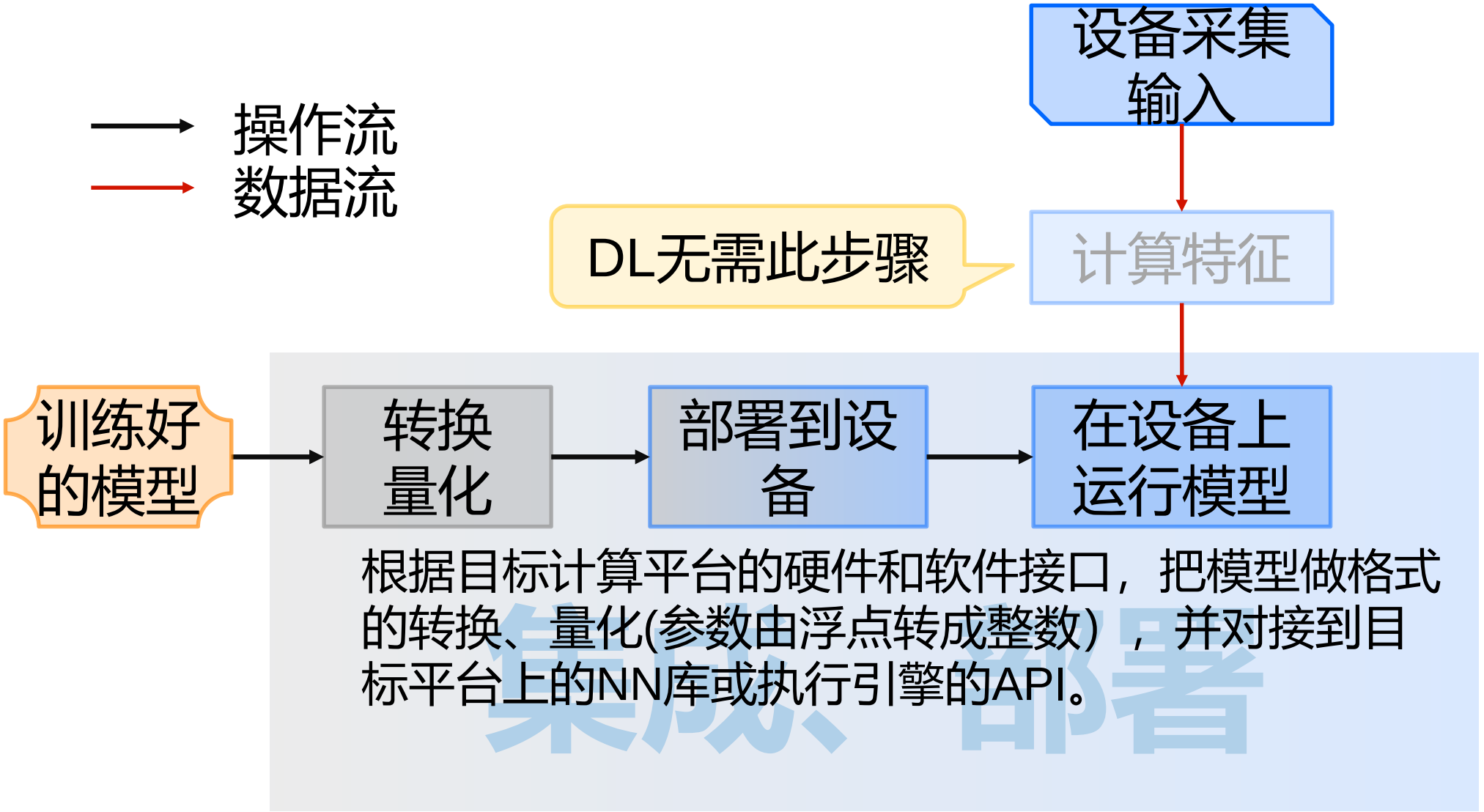
# AI为MCU应用“属性加成”

- 强化图像分类性能
  - 智能电器
  - 工业次品检测
  - 智能家居
- 强化音频分类性能
  - 口令识别
- 改进控制算法的性能
  - 电机与传动控制
  - 电源转换
  - 运动控制
- 异常状态与故障检测
  - 部件损坏
  - 过程稳定性
  - 意外：跌落、碰撞
  - 生理信号

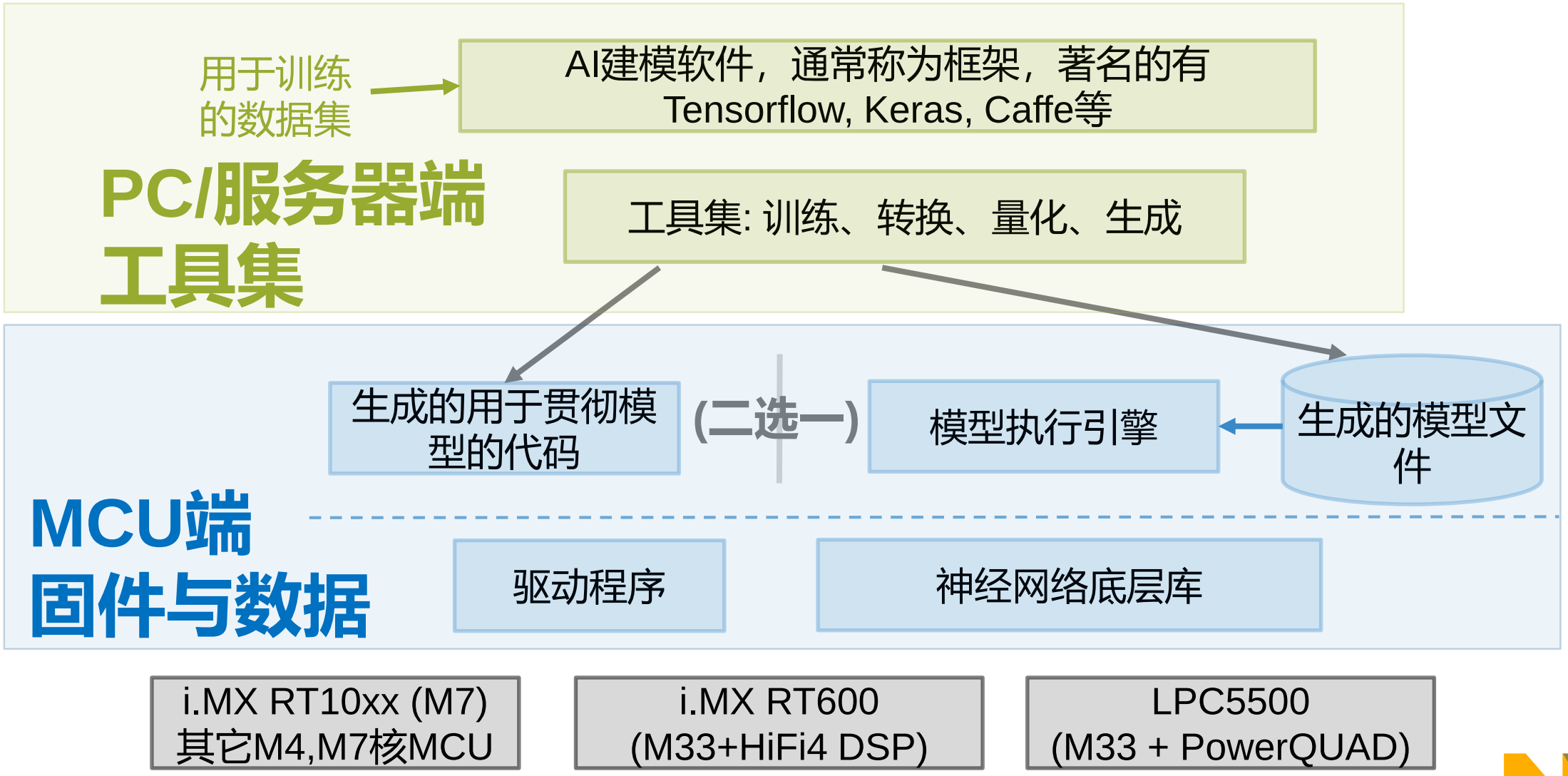
# 在MCU上集成AI的整体流程



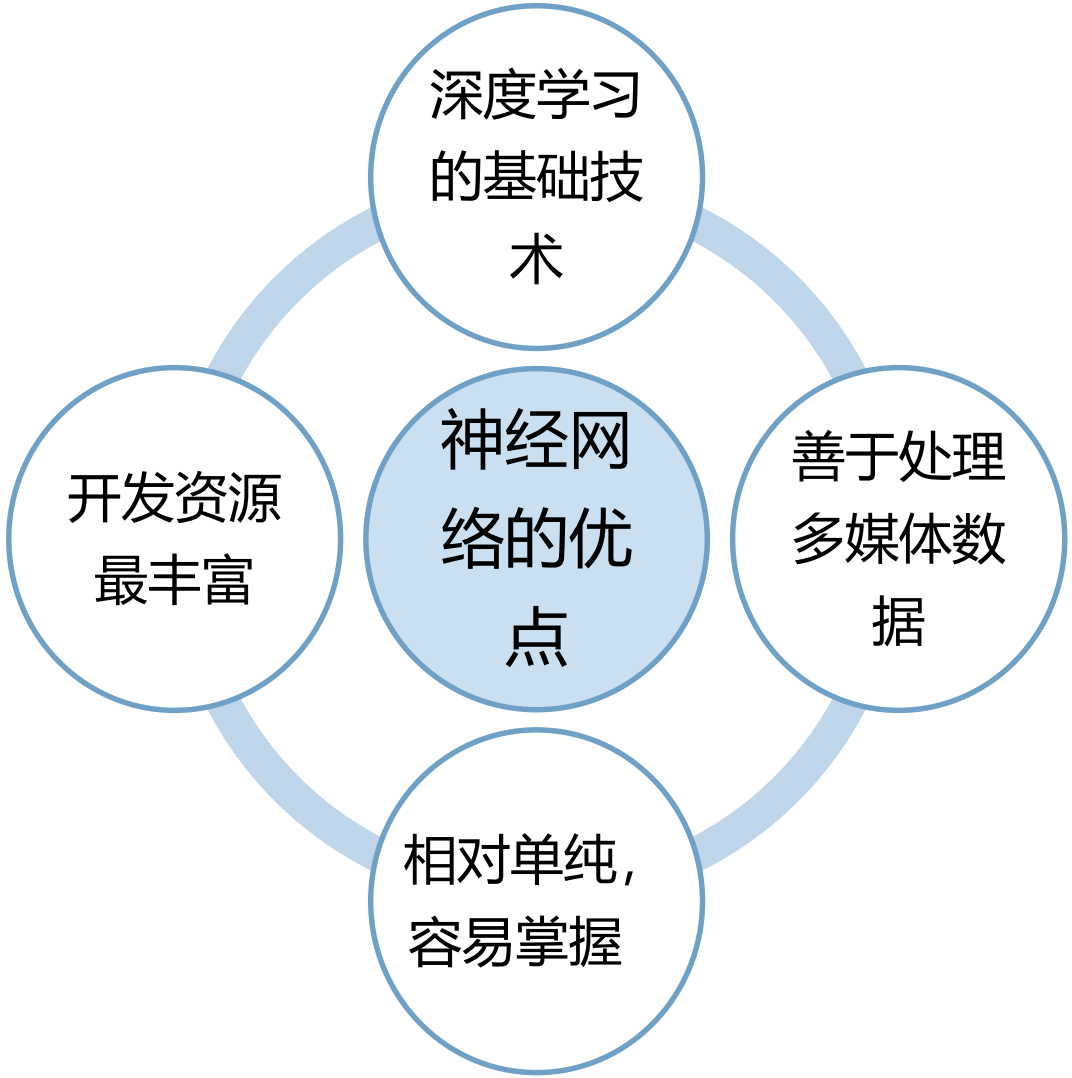
# 部署过程


操作流
数据流
设备采集输入
DL无需此步骤
计算特征
训练好的模型
转换量化
部署到设备
在设备上运行模型
根据目标计算平台的硬件和软件接口，把模型做格式的转换、量化(参数由浮点转成整数)，并对接到目标平台上的NN库或执行引擎的API。

# MCU上部署AI的配套工具

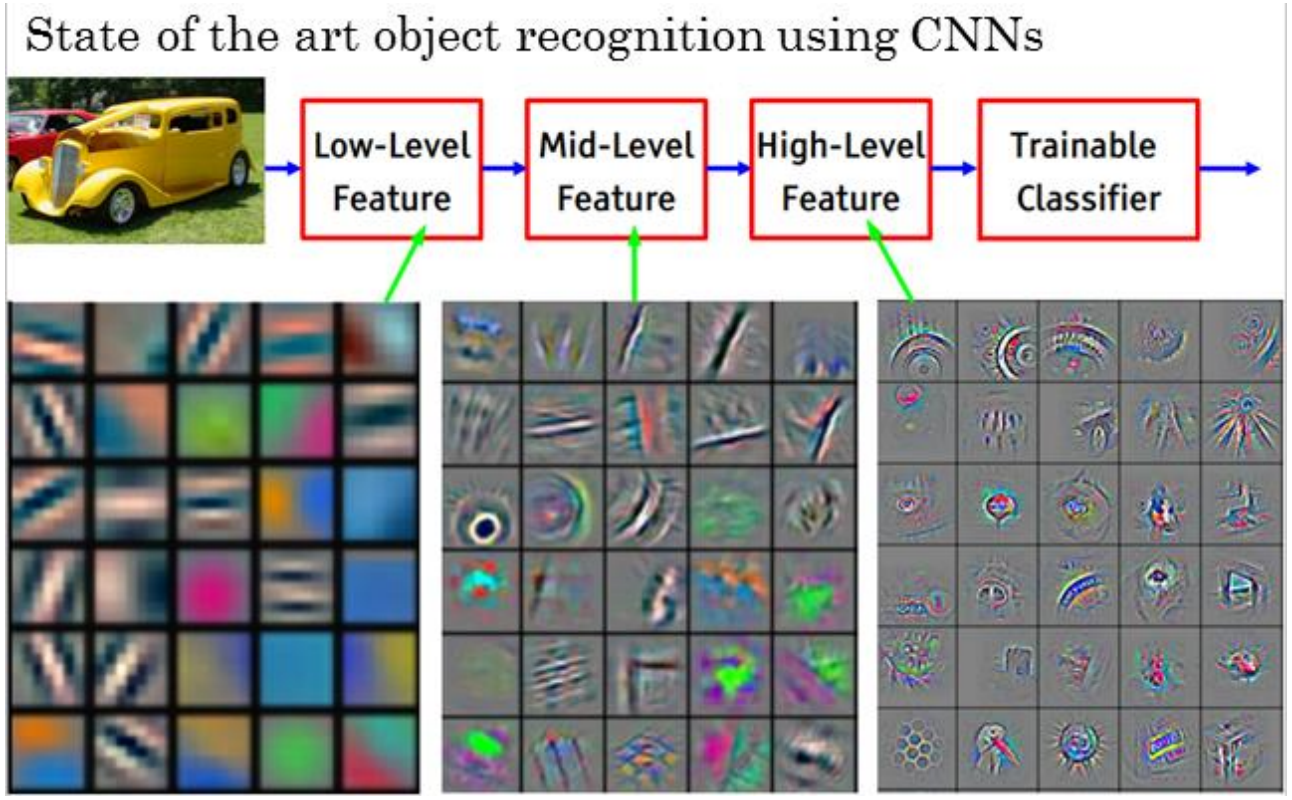


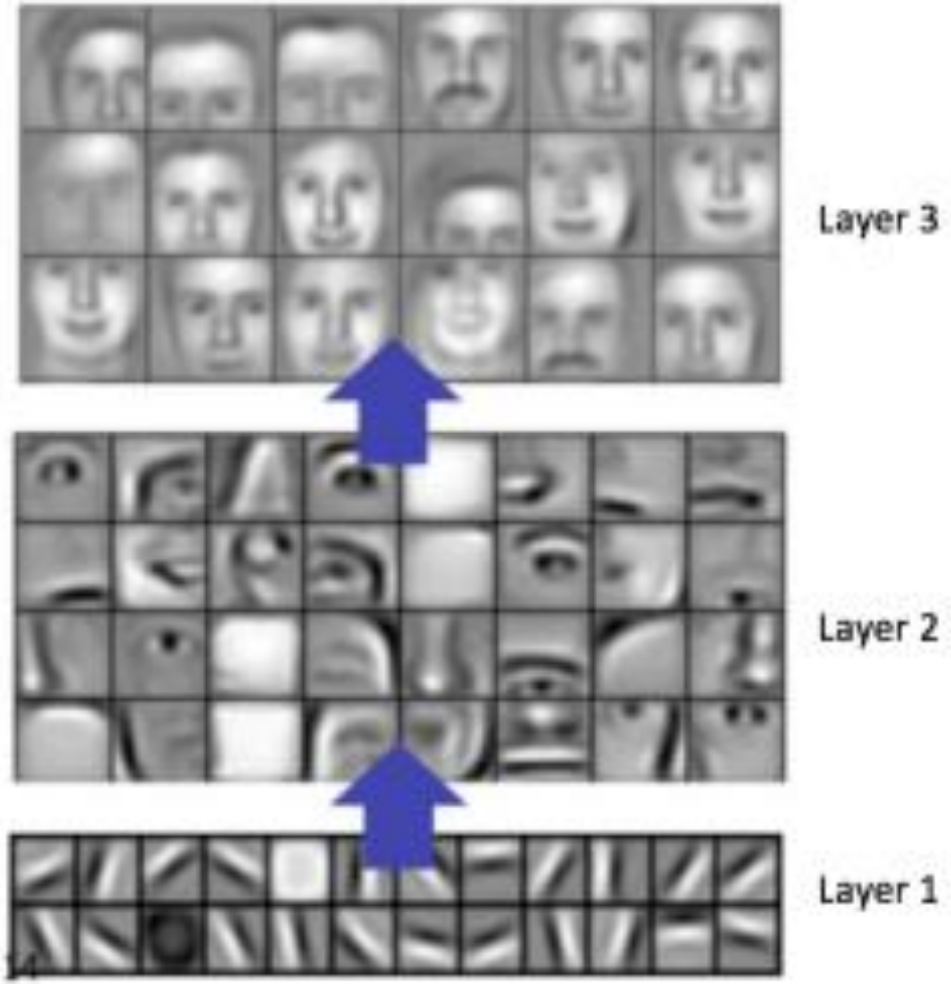
# MCU上基于神经网络(NN)建模的优点



# 深度神经网络工作原理模式图

- 提取初级特征：边角、线条等
- 提取中级特征：部件的轮廓
- 提取高级特征：部件
- 根据不同类别中各高级特征的重要性分类



# 适合在MCU端使用的NN基本构建块

## 主运算部

- 常规卷积层 (CNN)
- CNN按空间与通道分解 (DS-CNN)
- 全连接层 (Dense/FC/IP)

基本构建块

先做主运算

常常再做后加工

## (多数情况)后加工

- 激活 - 表达非线性关系
- 下采样/池化 - 精炼特征
- Softmax - 份量转成概率

# 构建块的常用搭建方式

## 串联/直筒式：一层一层往上迭

- 最简单的结构，如同“糖葫芦”。微型/小型网络的首选

## 直筒 + 跳跃：某层的输出又加到后续几层后

- 胜任较复杂的问题，如人脸识别。如MobileNet V2

## 化整为零: 大型构建块做并联/串联分解

- 处理大型问题，在MCU上耗时较长。如各代inception

## 预制复合结构

- 若干层构建块组成一个复合单元，后者再串在一起: 简化设计，灵活多用

# MCU端运行神经网络的基础软件

## CMSIS-NN

- 仅是底层NN库，需另行生成上层代码或提供执行引擎
- 针对Cortex-M高度优化
- 截止至2018.11仅支持CNN, DS-CNN, FC层
- 不支持并联结构
- 仅用于Cortex-M

## TensorFlow Lite

- 自带执行引擎和底层NN库
- 性能远不如CMSIS-NN
- 支持丰富的神经网络构建块与搭建方式
- 支持多种嵌入式平台

## 其它(制作中)

- HiFi4
- PowerQUAD
- 专用加速引擎

# 人工智能在机器视觉上的示例

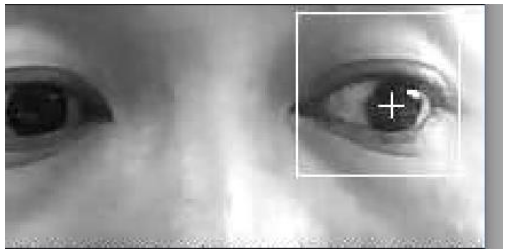
# 机器视觉的体验平台 – OpenMV RT

- OpenMV：开源的可编程摄像头项目
  - 以MIT许可证开源了软件、固件、硬件
  - 支持众多机器视觉功能和卷积神经网络执行引擎
  - 可编程：集成了micropython环境，并把功能包装成Python模块和类
  - 专用IDE: 开发Python应用脚本，实时监控处理结果

- OpenMV RT是在i.MX RT1050/60上的移植
  - 大幅提升算法性能
  - 接口与官方兼容，重用官方开发工具

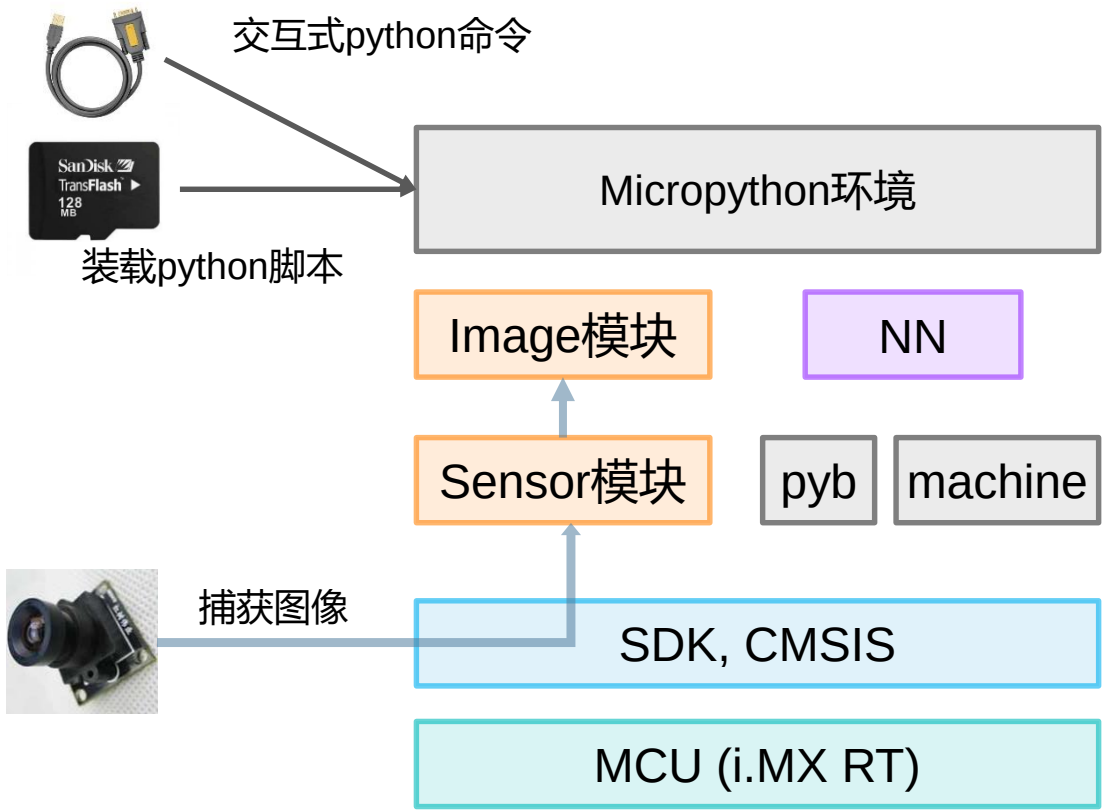
# OpenMV 的主要功能

- 使用Python编程MCU
- 基本绘图
- 图像采集与视频录制
- 数字图像处理
- 特征检测
- 脸部检测
- 色块检测
- 条码/二维码检测
- **神经网络执行引擎**

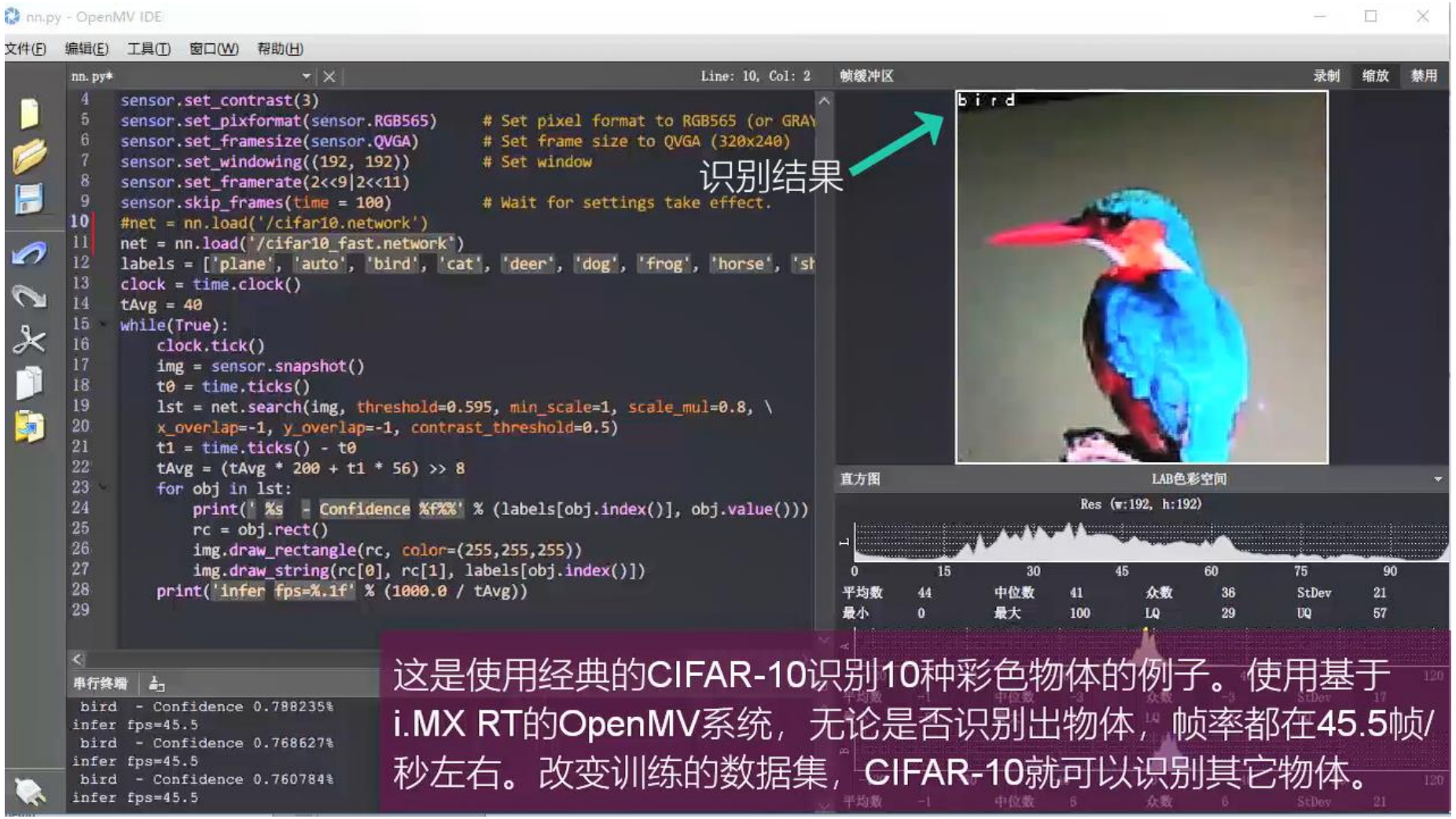
# OpenMV RT + CNN执行平台

- 使用Micropython开发应用，包含以下关键Python模块
  - (micropython & pyboard 标准 API) **pyb,machine**: 控制MCU通用资源
  - (openMV) **sensor**: 控制摄像头
  - (openMV) **Image** : openMV核心APIs
  - (OpenMV) **NN**: 神经网络模块.
- NN模块读取并执行SD卡中的模型，输入数据来自摄像头。支持串行CNN模型。



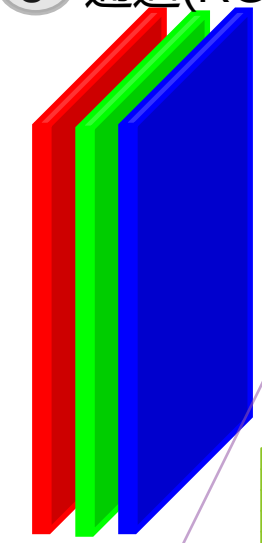

mpy porting repository
https://github.com/RockySong/micropython-rocky

# OpenMV RT上 “CIFAR-10”十项物体分类CNN演示



# CIFAR-10示例的结构

32x32
3 通道(RGB)

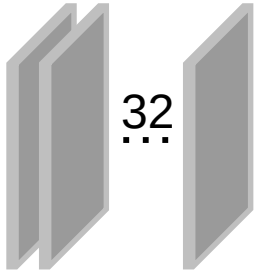


卷积核大小: 5x5

3 x 32
=96 个
卷积核

第1级:
卷积层 +
激活 +
向下采样

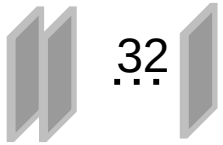
16x16
32 个
初级特征组




32 x 32
= 1K 个
卷积核

第2级:
卷积层 +
激活 +
向下采样

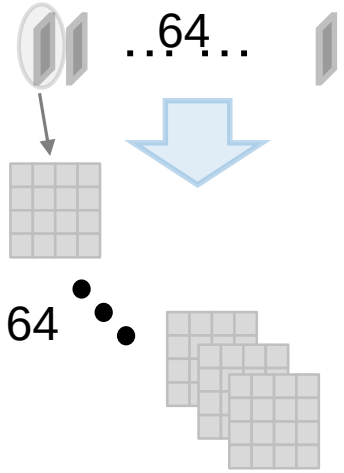
8x8
32 个
中级特征组




32 x 64
=2K 个
kernels

第3级:
卷积层 +
激活 +
向下采样

4x4
64 个
高级特征组



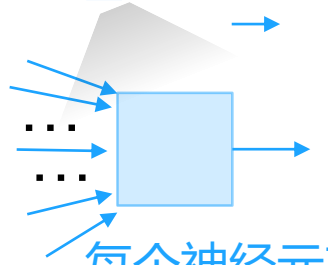

第4级: 全连接分类

$W_{10,1024} * X_{1024,1} + B_{10,1}$
矩阵乘向量

展平到1024
维特征向量


1024*10
=10K 权重

置信度

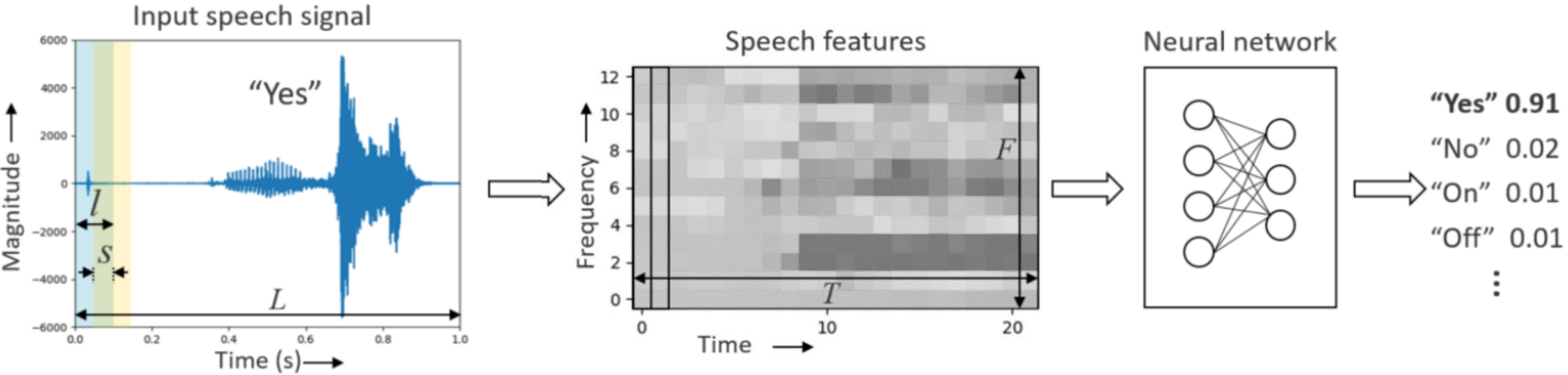
每个神经元有
1024 个权重

网络大小= (96+1K+2K)*  + 10K + 10 = 89,450=**87.3kB**

M7 MAC算力需求~= **13MOPS**

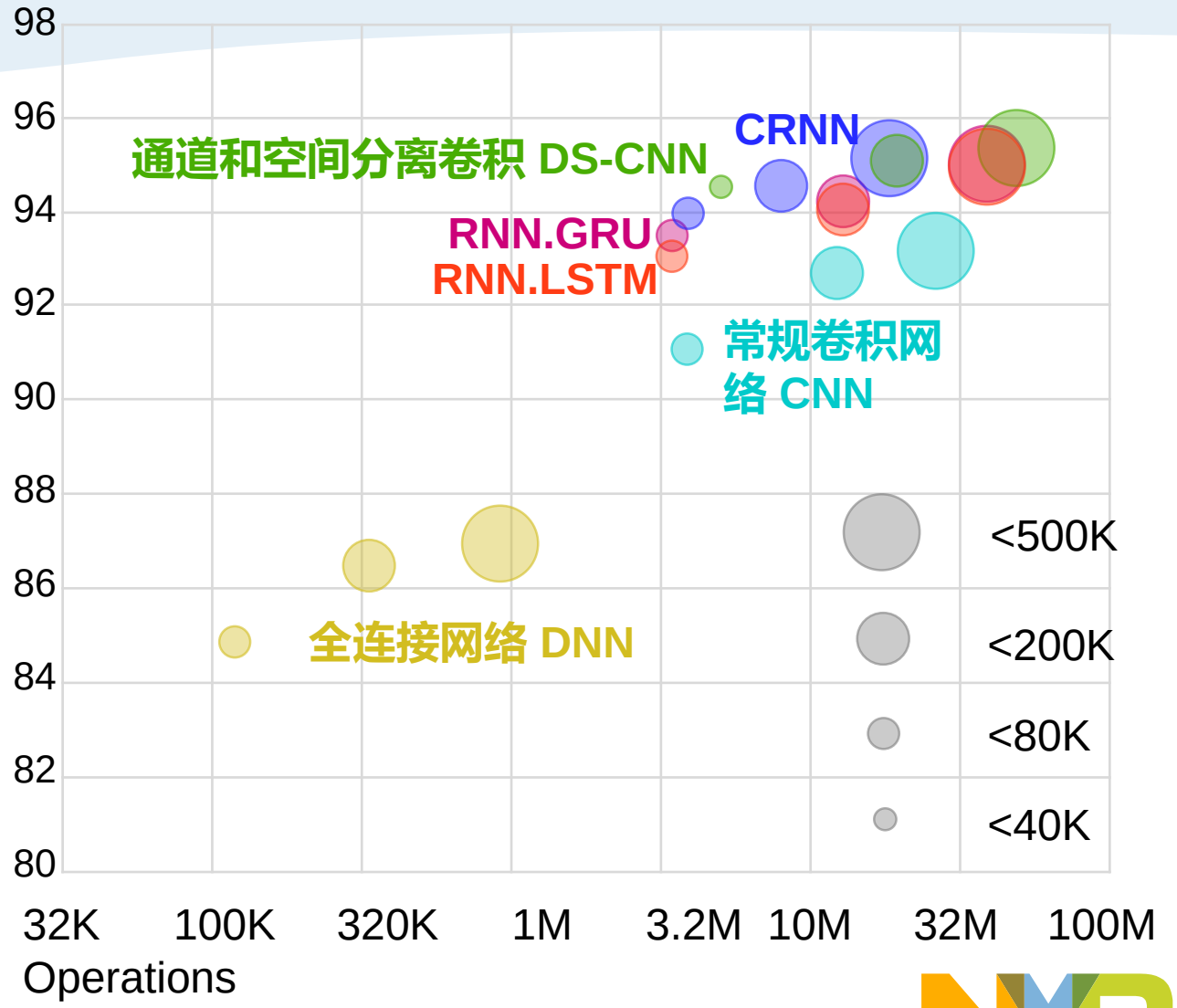
# 使用DL实现话音口令检测 (KWS)

- 接收短口令，并映射到预定义的命令
- 广泛用于语音控制的系统中，作为永久在线的低功耗唤醒模块
- 把时域信号分割变换成多段频谱，形成灰度图，再应用NN技术



# 关键词检测模型的资源需求

- 显著的边际效应递减
- 全连接神经网络(DNN) 精度最低，但计算量也最低.
- 常规CNN性能稍有不足
- **DS-CNN**有效减少了模型尺寸和提高性能，只是对算力要求稍有提高.
- RNN比常规CNN效果好, CRNN融合了CNN与RNN的优点.



# 小结

- 人工智能可以提升嵌入式系统的价值
- 在MCU上应用人工智能有独特的优点
- 神经网络是在MCU上实现人工智能的有效技术
- OpenMV RT可以在MCU上体验机器视觉和人工智能

SECURE CONNECTIONS
FOR A SMARTER WORLD