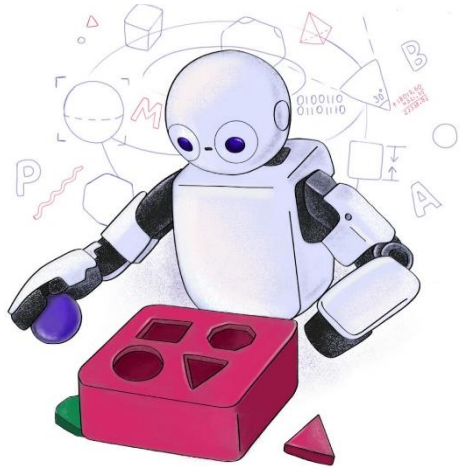
TP557 - Tópicos avançados em IoT e Machine Learning:

Introdução ao curso



Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Motivação: Modelos mais eficientes

- Os avanços em IA têm levado à criação de grandes modelos (como GPTs, DALL-E, Deep Seek, GroudingDINO, YOLOs entre outros), também chamados de *foundational models*.
- Esses modelos são usados para tradução de texto, assistência pessoal, detecção de objetos, geração de imagens sintéticas, etc.

Motivação: Modelos mais eficientes

deepseek

YOLO

- Para que tenham um altíssimo desempenho, esses modelos são treinados em bases de dados massivas.
- Além disso, devido à sua complexidade e ao volume de parâmetros, eles demandam enormes recursos computacionais – memória, armazenamento e energia – para treinamento e execução (também chamada de inferência).

Motivação: Modelos mais eficientes

- Por exigirem recursos computacionais colossais, isso os torna proibitivamente caros.
- Como resultado, a maioria dos usuários os acessa como serviços baseados em nuvem.
- O uso na nuvem resulta em maior latência, alto consumo de energia e custos operacionais elevados.

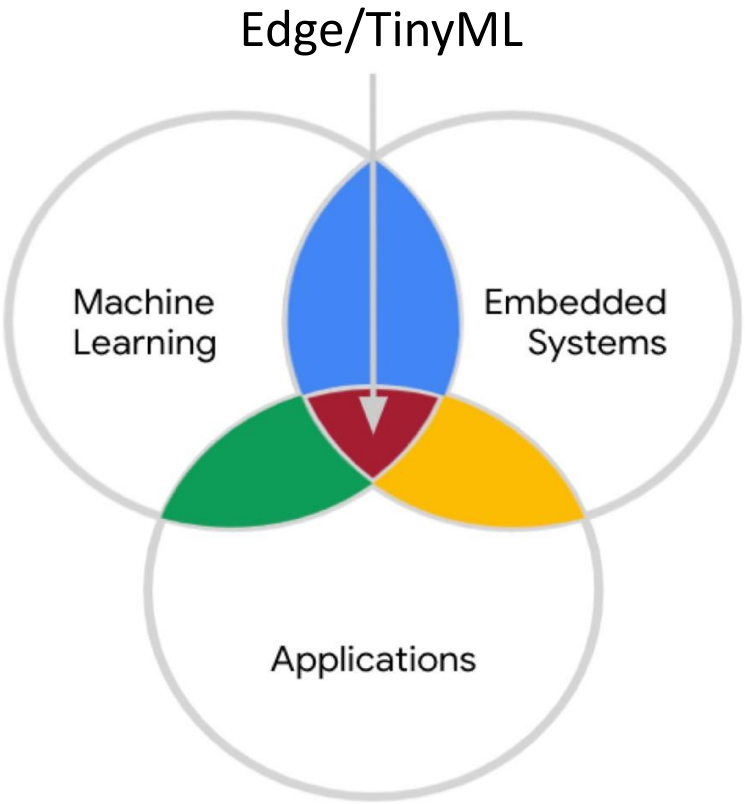
Motivação: Modelos mais eficientes

- Essa realidade torna inviável o uso desses e outros modelos em dispositivos com recursos limitados, como smartphones, *wearables* ou dispositivos IoT.

Motivação: Modelos mais eficientes

- Esses dispositivos são caracterizados por:
  - **Baixa capacidade de processamento:** microcontroladores ou processadores com poder de computação reduzido.
  - **Memória restrita:** pouca memória RAM e não volátil.
  - **Baixo consumo de energia:** devem ser energeticamente eficientes, sendo muitas vezes alimentados por baterias.
  - **Recursos de hardware simples:** dispõem de periféricos básicos e conectividade limitada.

Motivação: Modelos mais eficientes



- Nesse contexto, surgem os paradigmas edge e tinyML.
- Eles buscam adaptar técnicas de machine learning para ambientes restritos, permitindo a implementação de modelos otimizados e eficientes que podem ser executados localmente com baixo consumo de energia e memória.
  - O tinyML também é conhecido como embeddedML.

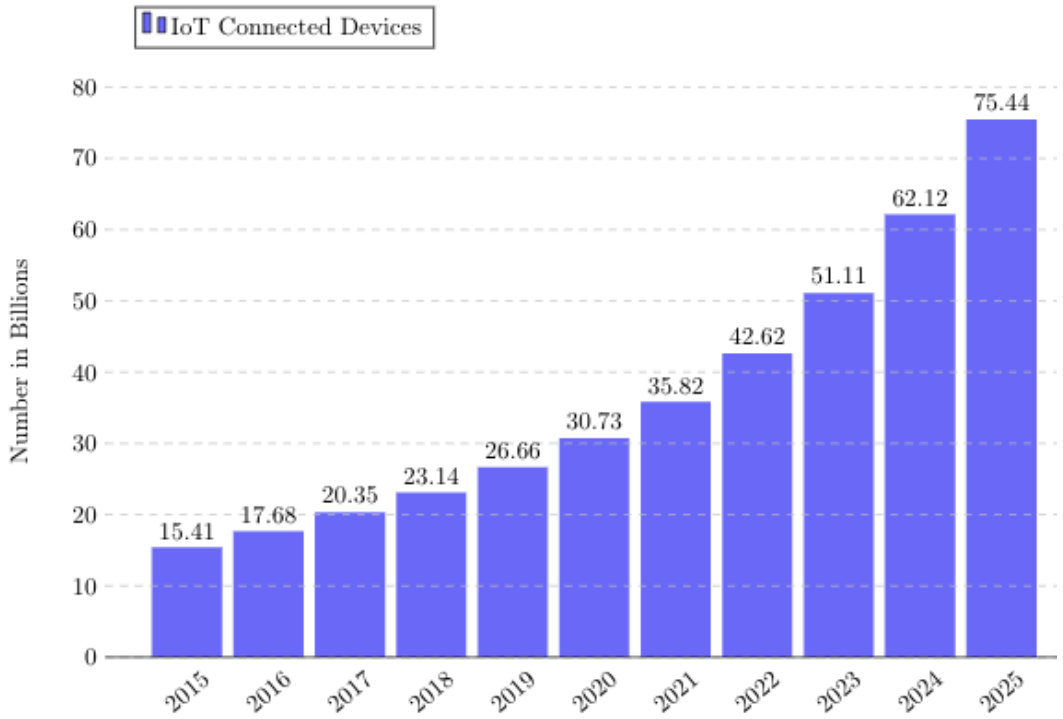
Objetivos do curso

- Apresentar os principais algoritmos e técnicas de aprendizado de máquina (em inglês, Machine Learning, ML) aplicados a dispositivos com recursos computacionais limitados.
- O curso terá duas partes, uma introdutória e outra prática.
- A primeira parte conterá aulas introdutórias sobre dispositivos embarcados, IoT e ML.
- A segunda parte conterá laboratórios e estudos dirigidos para fixação dos conceitos introduzidos, além de um projeto final.
- Ao final do curso, vocês deverão ser capazes de entender e aplicar os principais algoritmos e técnicas de aprendizado de máquina em aplicações práticas utilizando dispositivos com restrições computacionais.

Teste de Python

- Grande parte do curso necessitará de conhecimentos sobre a linguagem de programação Python, os quais eu assumo que vocês já têm.
- Assim, como primeira atividade, peço que vocês resolvam alguns exercícios sobre programação Python como forma de avaliar seus conhecimentos.
- Caso vocês sintam dificuldades em resolvê-los, me procurem, pois posso sugerir cursos e livros.
- **Estes exercícios valem nota.**
- O link para resolução dos exercícios via Google Colab segue abaixo:
  - https://colab.research.google.com/github/zz4fap/tp557-iot-ml/blob/main/exercises/Exerc%C3%ADcios_sobre_Programa%C3%A7%C3%A3o_em_Python.ipynb

Motivação: Quantidade



- Estima-se que teremos ***mais de 75 bilhões de dispositivos*** IoT até 2025.
- A grande maioria desses dispositivos estão ***equipados com microprocessadores*** ou ***microcontroladores,*** (vários) ***sensores*** e ***atuadores***.

Motivação: Economia

NVIDIA Ampere A100
Graphics Processing
Unit (GPU)
400 Watts
826 mm^2

Apple A12 (iPhone)
System on a chip (SoC):
CPUs, GPU e Video Codec
3.64 Watts
83 mm^2

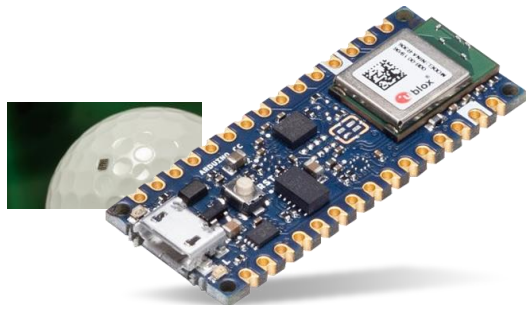
Apple APL0778 (iWatch)
Application Processor
Unit (APU): CPU+GPU
< 0.78 Watts
32 mm^2

Tiny

Syntiant NDP100
Neural Decision
Processor (NDP)
140 μWatts
2.52 mm^2

- Se comparados com servidores, equipados com dezenas de CPUs e GPUs, os processadores e microcontroladores desses dispositivos são muito ***menores*** e consomem muito ***menos energia*** e são ***mais baratos***, devido a alta demanda.

Motivação: Baixo consumo de energia



- O baixíssimo consumo de energia possibilita que os dispositivos:
  - ***Funcionem por longos períodos*** de tempo sem trocar a bateria (e.g., anos);
  - E executem aplicações "***Always On***":
    - ✓ Aplicação constantemente ativa e monitorando o ambiente ao seu redor, coletando informações, as processando e respondendo em tempo real, sem interrupção.
    - ✓ Característica muito importante em aplicações que exigem monitoramento constante, como sistemas de ***segurança***, monitoramento de ***saúde***, sistemas de ***automação industrial*** ou dispositivos de ***assistência virtual***.

https://youtube.com/shorts/O8cmpRZ9dOo?feature=share

Motivação: Sensores

- Dispositivos IoT, também chamados de ***endpoints***, possuem muitas vezes uma grande variedade de sensores!

Sensores de movimento
Giroscópios, radares (e.g., LiDAR), magnetômetros, acelerômetros.

Sensores de acústicos
Microfones, ultrassônicos, vibrômetros, geofones.

Sensores ambientais
Temperatura, humidade, pressão, infra-vermelho.

Sensores de toque
Capacitivos, Resistivos.

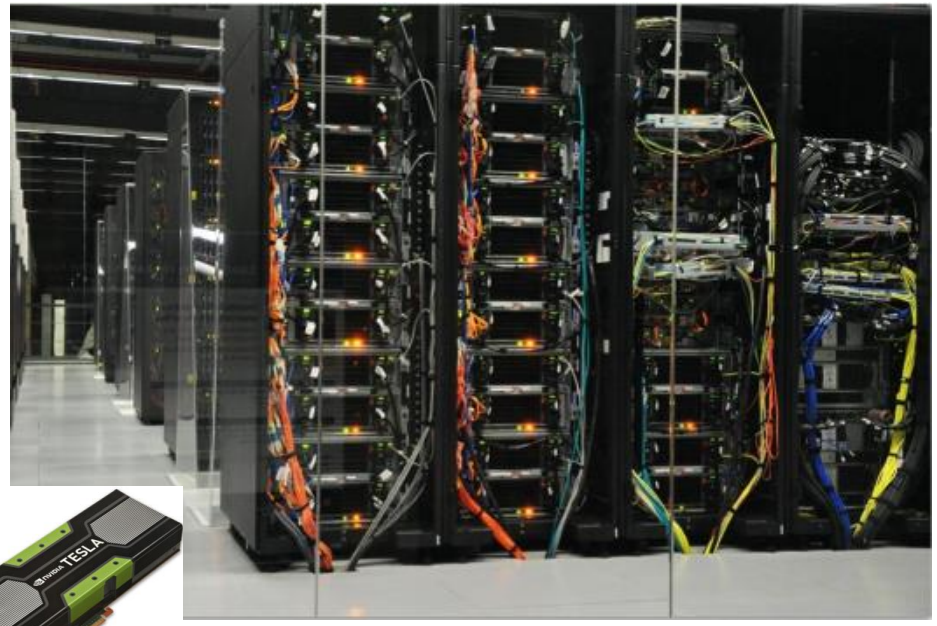
Sensores de imagem
RGB, Térmicos, LiDAR.

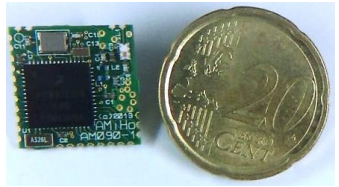
Sensores biométricos
impressão digital, batimento cardíaco, glicose.

Sensores de força
Pressão, deformação/tensão.

Sensores de rotação
Encoders, potenciômetros.

Motivação: *Always On* ML





E se nós pudéssemos executar aplicações Always-On baseadas em inteligência artificial (IA), mais especificamente machine learning (ML), nesses dispositivos?

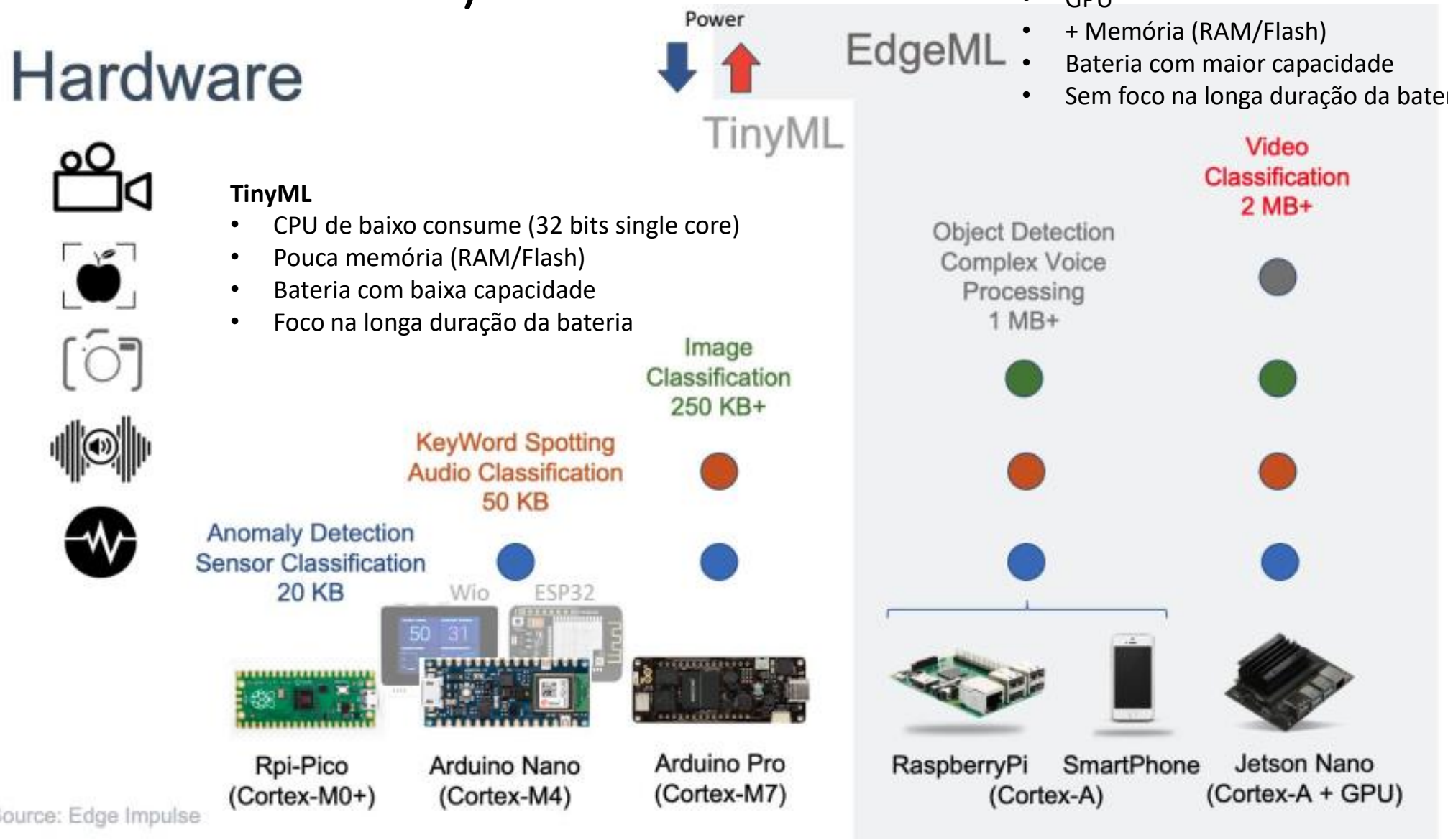
O que poderíamos fazer?

EdgeML versus TinyML

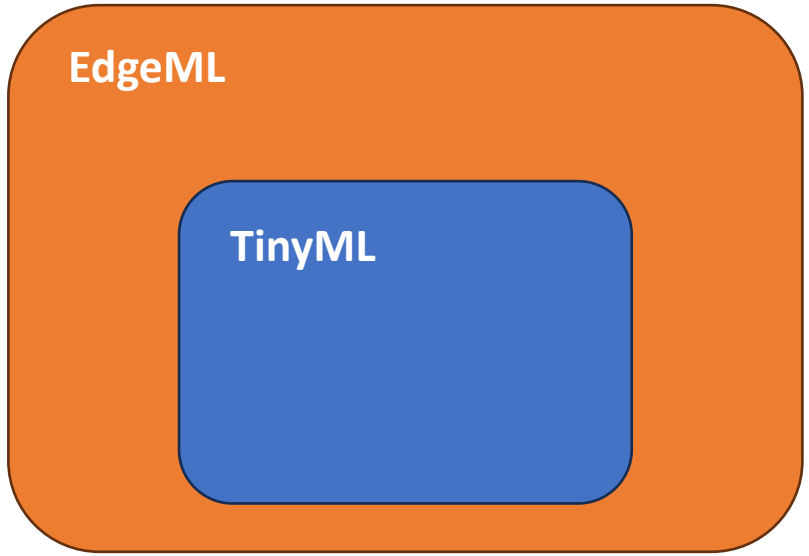
Diferenciação de hardware e terminologia

TinyML
- CPU de baixo consume (32 bits single core)
- Pouca memória (RAM/Flash)
- Bateria com baixa capacidade
- Foco na longa duração da bateria

EdgeML
- CPU mais poderosa (64 bits e 2+ cores)
- GPU
- + Memória (RAM/Flash)
- Bateria com maior capacidade
- Sem foco na longa duração da bateria



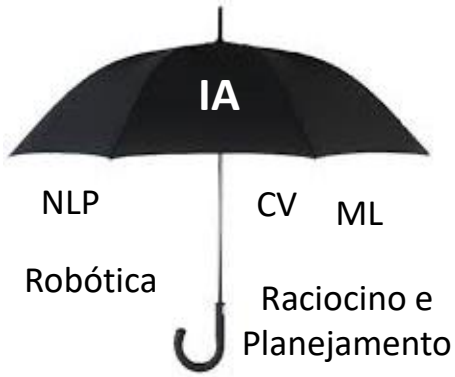
Edge Computing versus EdgeML versus TinyML

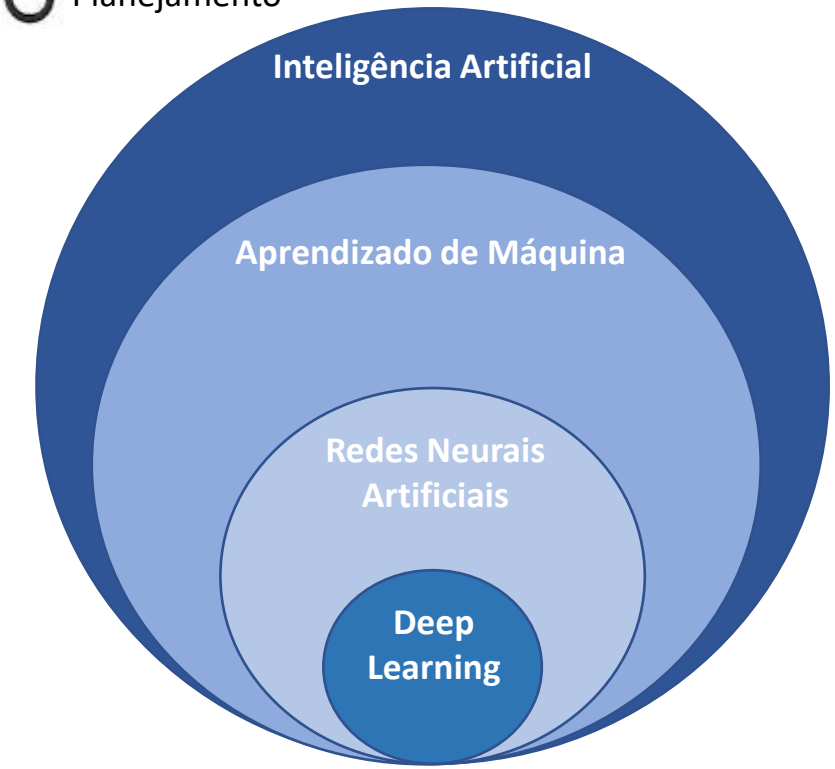


Nosso curso focará no uso de dispositivos tinyML.

- **Edge Computing**: processamento e armazenamento de dados é feito ***próximo ou nos dispositivos que coletam os dados*** (ou seja, na borda da rede) e não na nuvem.
- **EdgeML**: processamento de algoritmos de IA na borda, ou seja, nos dispositivos e sensores.
- **TinyML**: subconjunto do ***EdgeML***, onde os dispositivos coletam e processam dados com ***consumo de energia ultrabaixo*** (baterias), possibilitando ***a execução contínua de modelos de ML*** (i.e., dispositivos "***always on***").

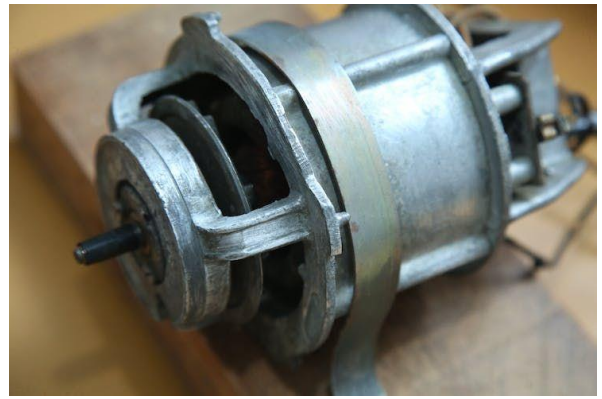
Mas o que são IA e ML?





- **IA**: técnicas que façam as ***máquinas imitarem o comportamento humano***.
- **ML**: algoritmos que ***aprendem uma tarefa sem serem explicitamente programados para tal***.
- **RNAs**: modelos ***inspirados no funcionamento do cérebro humano***. Consistem em neurônios artificiais interconectados, que são organizados em camadas e capazes de aprender a partir de experiências prévias.
- **DL**: modelos que extraem (i.e., aprendem) padrões complexos de dados usando redes neurais com várias camadas (i.e., profundas).
  - Por terem ***maior capacidade***, ***necessitam de grandes quantidades de dados*** para aprender.

Aplicações do tinyML



Manutenção preventiva

- **Aplicação**: ***monitoramento contínuo*** de equipamentos com o objetivo de ***prever falhas***.
- **Sensores que podem ser utilizados**: movimento, corrente, áudio, câmera.

Aplicações do tinyML

Monitoramento e rastreio de bens

- **Aplicações**:
  - Monitoramento da saúde de animais e lavouras.
  - Detecção de doenças em plantações.
  - Detecção de incêndios.

- **Sensores que podem ser utilizados**: movimento, temperatura, humidade, posição, acústico, câmera.

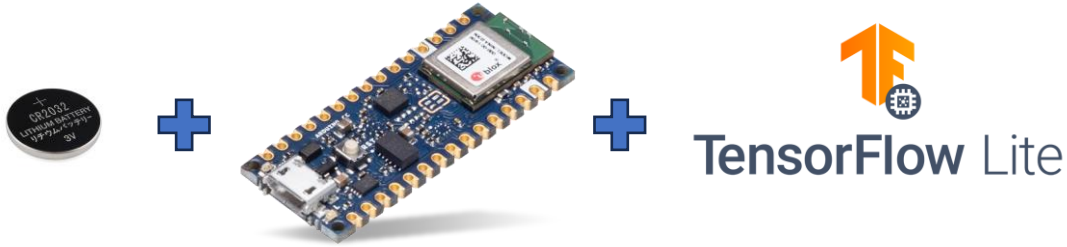
Aplicações do tinyML

E-health

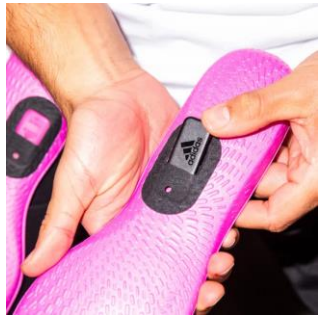
- **Aplicações**:
  - Detecção de fibrilação arterial em sinais eletrocardiograma.
  - Detecção de doenças pulmonares (e.g., COVID, pneumonia).
  - Detecção de apneia do sono.
  - Detecção de posturas incorretas.
  - Previsão de hospitalização por insuficiência cardíaca

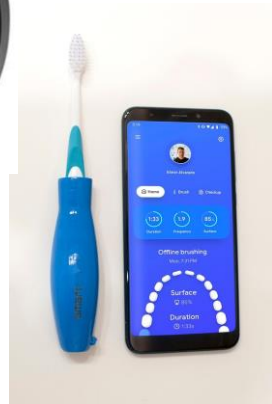
- **Sensores que podem ser utilizados**: movimento, acústico, imagem e ECG.

Sobre o curso

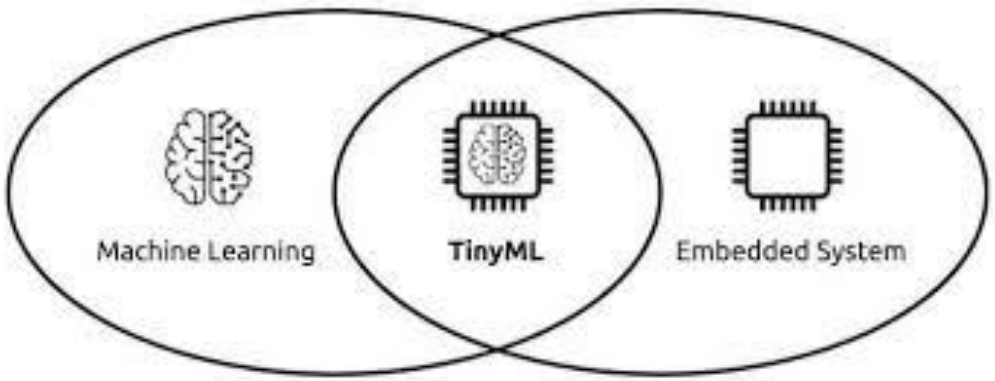


- A difusão de dispositivos de ***(ultra)baixo consumo de energia***, equipados com sensores (e atuadores), juntamente com a introdução de ***bibliotecas de IA para microcontroladores***, têm permitido a proliferação em massa de dispositivos de ***Internet das Coisas*** (IoT) ***Inteligentes***, ou seja, que executam algoritmos de ML.
- Esse novo paradigma é chamado de ***TinyML***: ***IoT + Sensores + IA (ML)***.



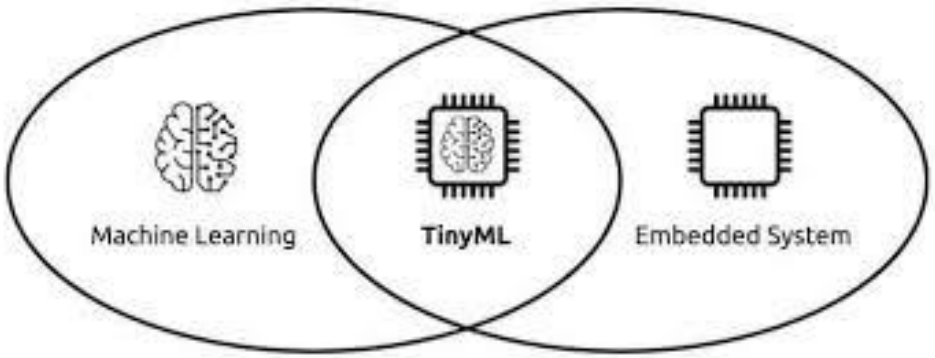


Sobre o curso



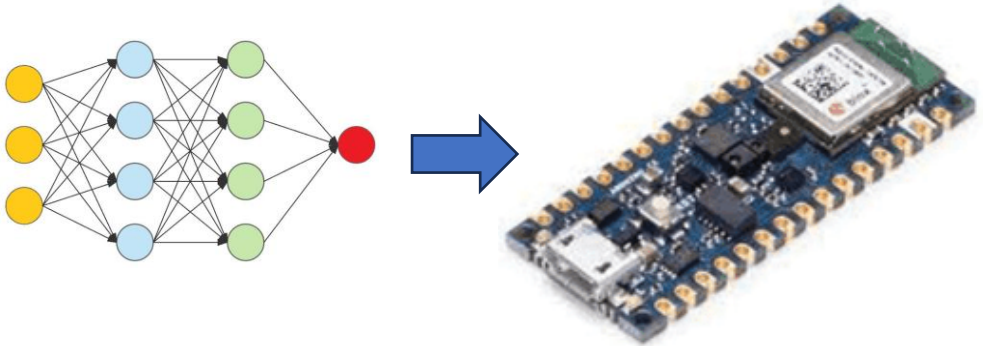
- **TinyML**: subárea da IA que lida com a ***implementação de algoritmos e ferramentas otimizadas para a execução de modelos de ML em dispositivos de baixo consumo de energia***, como como sensores IoT, *wearables* e dispositivos portáteis.
- Envolve a ***minimização do tamanho do modelo***, o ***uso de algoritmos de ML leves*** e a ***otimização do uso dos recursos computacionais*** dos dispositivos.

Sobre o curso



- O ***processamento local*** dos dados em ***dispositivos de borda*** traz benefícios como:
  - ***Menor latência***: não é necessário o envio dos dados para um servidor na nuvem, possibilitando aplicações que operem em tempo real;
  - ***Maior privacidade***: dados sensíveis não são transferidos para servidores;
  - ***Menor dependência de conexões***: reduz ou elimina a necessidade de transferência de dados para servidores.
  - ***Maior autonomia energética***: o uso de microcontroladores com baixo consumo de energia reduz o consumo de energia, uso de recursos e custos.
  - ***Reduzido consumo de memória***: modelo e binário (i.e., executável) menores.

Sobre o curso



- Portanto, o curso mistura ML com dispositivos IoT, cujas principais características são:
  - **Baixo poder computacional**: geralmente equipados com CPUs single core de 32 bits, poucos quilobytes de memória RAM e alguns megabytes de memória de longo prazo (*flash*).
  - **Baixíssimo consumo de energia:** são geralmente alimentados à bateria, as quais devem durar um longo período de tempo.
  - **Sensores**: temperatura, pressão, humidade, ópticos (e.g., câmeras RGB, térmicas), movimento (e.g., giroscópio e acelerômetro), biométricos (e.g., impressão digital, batimento cardíaco), etc.
- Nosso curso é baseado nos cursos de ***TinyML*** de Harvard e da UNIFEI.
  - https://scholar.harvard.edu/vijay-janapa-reddi/classes/cs249r-tinyml
  - https://github.com/Mjrovai/UNIFEI-IESTI01-TinyML-2022.1

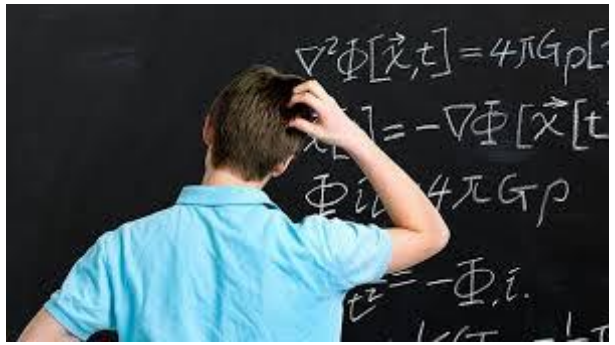
Sobre o curso

- O curso será dividido em duas partes:
  - **Fundamentos**: Introdução e desafios do TinyML, paradigma do ML, introdução ao *Deep Learning*, modelos de DL para regressão e classificação, redes convolucionais, como evitar problemas de sobreajuste e introdução ao ***Edge Impulse***;
  - **Aplicação prática dos fundamentos**: ciclo de vida e de trabalho de aplicações de ML, introdução às bibliotecas ***TFLite*** e ***TFLite-Micro***, visão geral do kit de desenvolvimento, laboratórios práticos usando o kit (ou celulares pessoais) e modelos de ML e apresentação de trabalhos práticos.
- **Pré-requisitos** (*não mandatórios*):
  - Conceitos básicos de ***álgebra linear*** (e.g., matrizes e vetores), ***cálculo*** (e.g., algoritmos de otimização, como o gradiente descendente), ***probabilidade e estatística*** (e.g., distribuições de probabilidade, média, desvio padrão, validação cruzada) e ***processamento de sinais*** (e.g., FFT, filtragem);
  - Conhecimentos básicos de programação em C/C++ e Python.

Sobre o curso

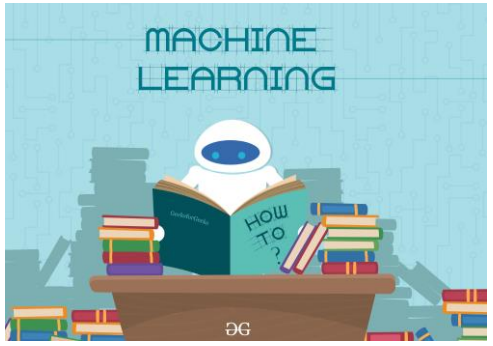
Avaliações

- Atividades (A): 30%
  - Exercícios de programação
  - Quizzes
  - Proposta de projeto final prático (ao final do segundo mês)
- Relatórios dos Laboratórios (R): 30%
  - Cinco (5) laboratórios com experimentos práticos.
- Projeto final (P): 40%
  - Projeto prático unindo IoT e ML.
  - Até dois (2) alunos por projeto.
  - Tema deve ser escolhido por vocês.

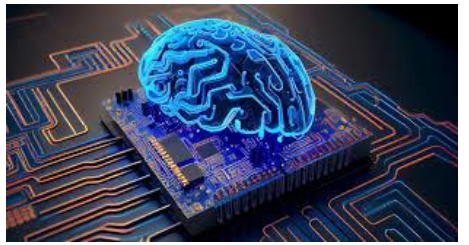


NF = A*30% + R*30% + P*40%

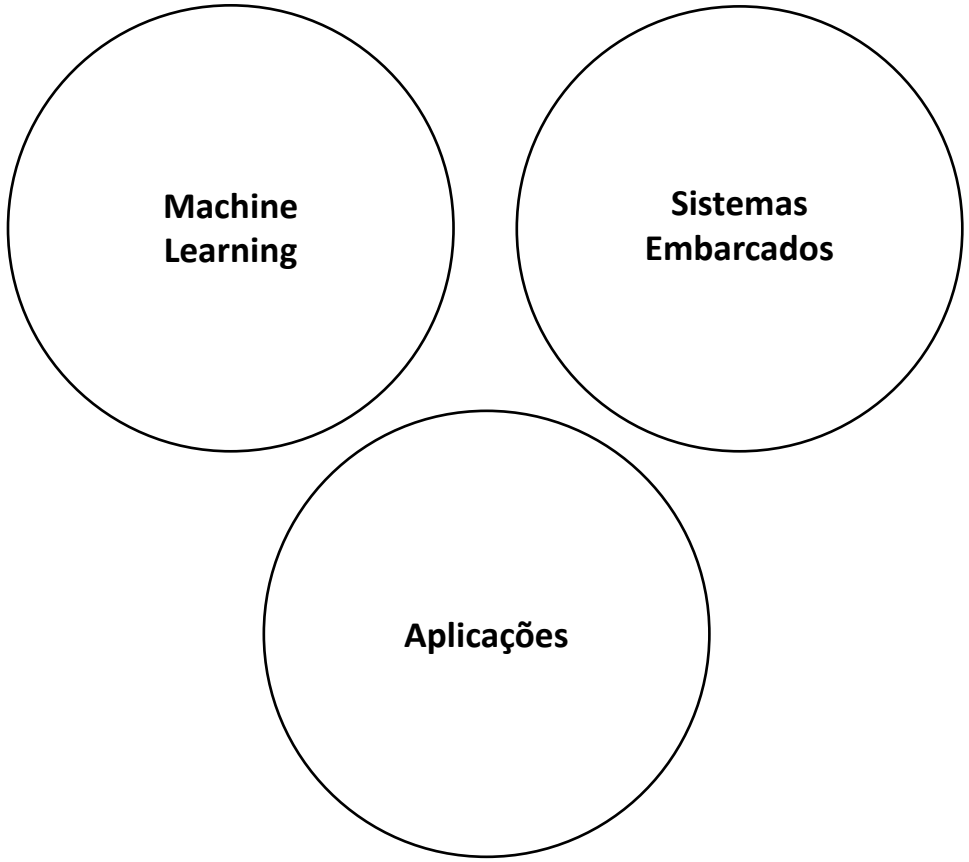
Objetivos do curso



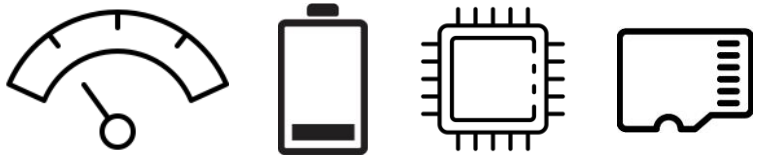
- Apresentar os ***fundamentos de IoT e ML***, bem como explorar a ***interseção*** entre essas áreas.
- ***Familiarizar*** os alunos com a ***literatura e os conceitos-chave da área de TinyML***.
- Capacitar os alunos a ***conceber***, ***treinar e implantar aplicações TinyML em sistemas embarcados*** (e.g., smartphones e dispositivos IoT).
- Proporcionar ***experiência prática por meio de laboratórios e projetos práticos***, permitindo que os alunos apliquem seus conhecimentos de TinyML em cenários do mundo real.



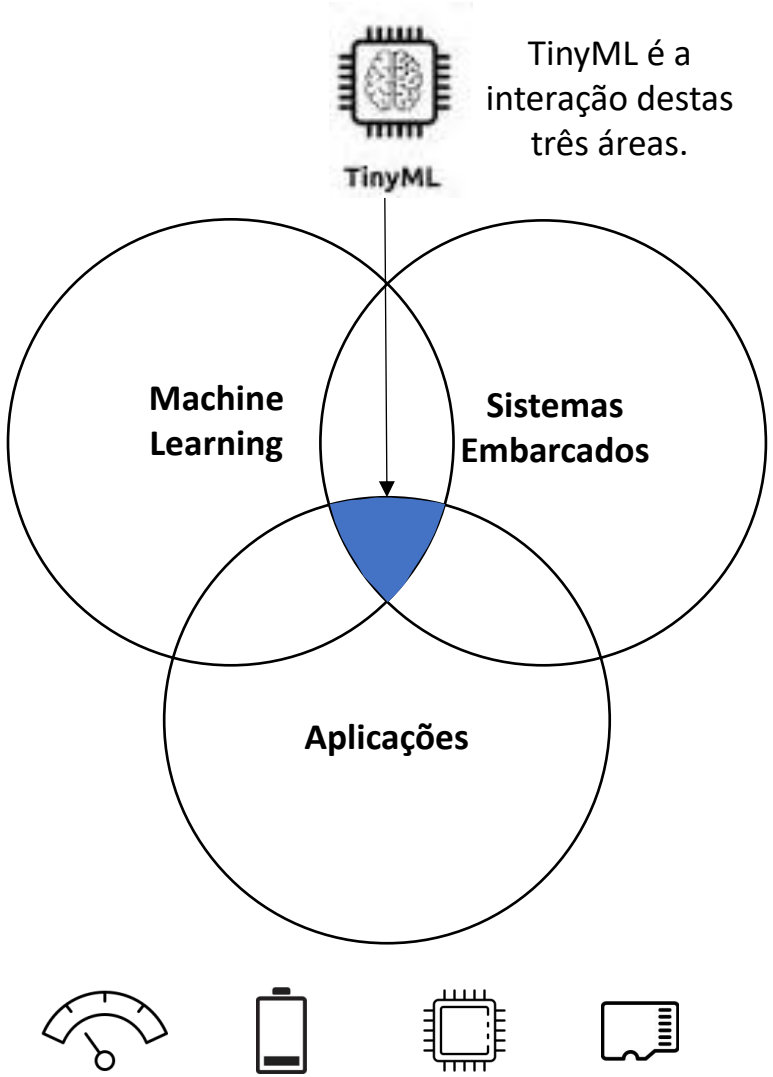
O que vamos aprender?



- Aprenderemos os ***fundamentos*** de cada uma dessas áreas, mas apenas o ***suficiente*** para focarmos na construção de ***aplicações*** que apresentem:
  - Baixa latência;
  - Maior autonomia energética;
  - Eficiência computacional;
  - Baixo consumo de memória.



O que vamos aprender?

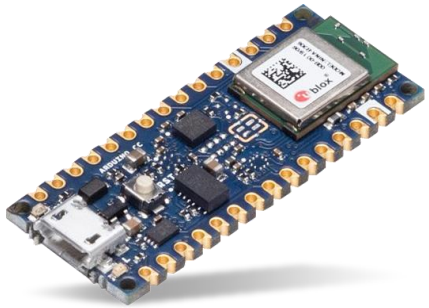


- Trabalharemos na ***interseção dessas três áreas***, que dá origem ao ***tinyML***.
- Para isso, devemos levar em consideração algumas questões importantes:
  - Como o ML pode ***viabilizar novas aplicações*** em dispositivos embarcados?
  - Quais são os ***desafios de implementar ML em dispositivos com recursos limitados***?
    - ✓ Temos que considerar problemas de ***latência*** (e.g., tempo para classificação de objetos em uma imagem) e restrições de ***consumo***, ***processamento*** e ***memória***.
  - Quais ***novos casos de uso podem ser habilitados com o uso de TinyML***, que não eram possíveis antes?

O que vamos usar?

Hardware

- Smartphones
- Arduino Nano 33 BLE Sense
- Sensores



Software

- **Bibliotecas de ML**: Tensorflow, TFLite e TFLite Micro.
- **Ambientes de programação**: Google Colab (com Jupyter notebooks) e Arduino IDE.
- **Ambiente de desenvolvimento de ML**: Edge Impulse Studio.

TensorFlow

jupyter

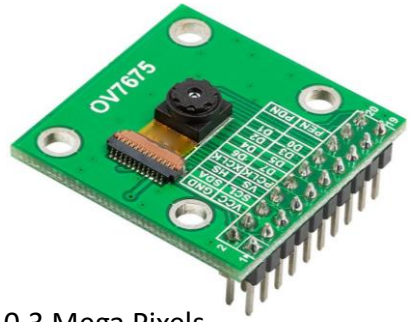
ARDUINO

EDGE IMPULSE

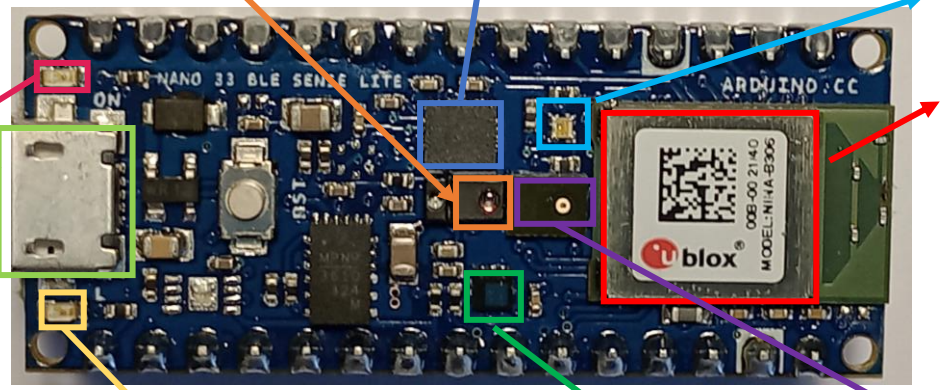
O kit TinyML

O kit contém:

- 1 x Arduino Nano 33 BLE Sense ***Lite****
- 1 x Shield
- 1 x Câmera OV7675
- 1 x Cabo USB A



Câmera de 0.3 Mega Pixels
Resoluções (pixels): 640x480 (VGA), 320x240 (QVGA) e 160x120 (QQVGA)
Frames por segundo: 30 (VGA), 60 (QVGA) e 240 (QQVGA)

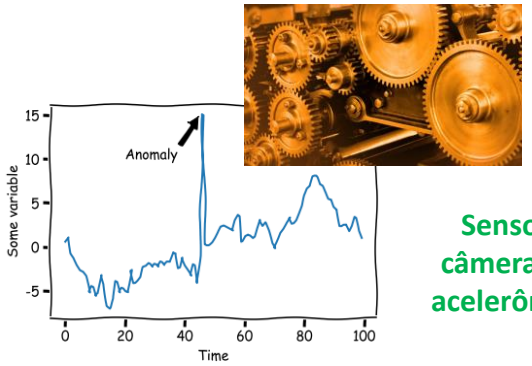


* Sem sensor de temperatura e humidade.

Atividades práticas

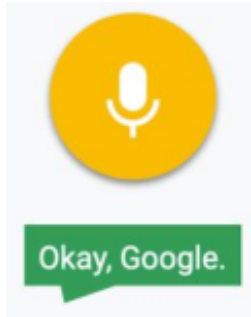
Classificação de movimento ou atividade

Sensores: acelerômetro, giroscópio, magnetômetro



Detecção/previsão de anomalias

Sensores: câmera e/ou acelerômetro.

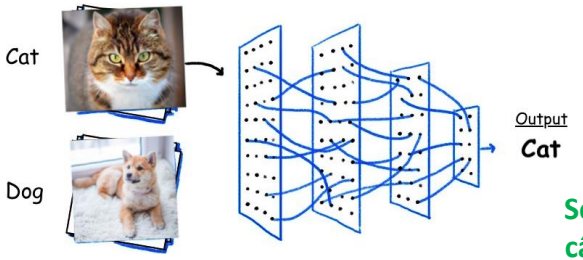


Keyword spotting

Sensor: microfone.

Visual wake words

Sensor: câmera.



Classificação de imagens

Sensor: câmera

Classificação de gestos

Sensores: câmera e/ou acelerômetro.

Referências

[1] Daniel Situnayake and Pete Warden, “*TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*”, 1st ed., O'Reilly Media, 2019.

[2] Aurélien Géron, “*Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*”, 1st ed., O'Reilly Media, 2017.

[3] Agus Kurniawan, “*IoT Projects with Arduino Nano 33 BLE Sense: Step-By-Step Projects for Beginners*”, Apress, 2020.

[4] Stuart Russell and Peter Norvig, “*Artificial Intelligence: A Modern Approach*”, Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. “*Deep Learning*”, MIT Press, 2016.

[6] Hakima Chaouchi, “The Internet of Things: Connecting Objects”, Wiley, 2010.

[7] Jun Zheng and Abbas Jamalipour, “Wireless Sensor Networks: A Networking Perspective”, Wiley, 2009.

[8] Coleção de livros

Avisos

- Todo material do curso está disponível no GitHub:
  - https://github.com/zz4fap/tp557-iot-ml
- Google Colab + Python Crash Course
  - https://www.youtube.com/watch?v=inN8seMm7UI&ab_channel=TensorFlow
  - https://www.youtube.com/watch?v=inN8seMm7UI
  - https://www.youtube.com/watch?v=pq4NNIYar9o&list=PLRc6ZYt68prVXAhwY1JD6DFc3BJGmJriq&pp=gAQBiAQB
- Introdução ao TensorFlow
  - https://www.youtube.com/watch?v=yjprpOoH5c8&ab_channel=TensorFlow
- Horário de Atendimento
  - Todas as quartas-feiras das 17:30 às 18:30.
  - Presencialmente ou remotamente.

Ideias para projetos finais

1. Identificação de falhas em rolamentos
2. Identificação de doenças respiratórias
3. Detectando falhas em automóveis através de seu som
4. Identificação de pragas
5. Identificação de posturas incorretas
6. Inspeção automática de qualidade
7. Classificador de Vogais em Libras
8. Detecção de doença pulmonar
9. Detecção de algarismos em hidrômetros
10. Detecção de Roncos

Ideias para projetos finais

1. Detecção de tumores cancerígenos
2. Detecção e prevenção de abalos sísmicos
3. Detecção de incêndios
4. Detecção de COVID através da tosse
5. Detecção do uso de máscaras
6. Personal Trainer
7. Predição da qualidade do ar
8. Diabetic retinopathy detector
9. https://www.tensorflow.org/lite/examples?hl=pt-br
10. https://experiments.withgoogle.com/collection/tfliteformicrocontrollers

Ideias para projetos finais

1. “Explore real-world edge ML projects”, http://tinyurl.com/tinymlprojects

Atividades

- Quiz: “***TP557 – Introdução***”
- Exercícios: Programação em Python.

Perguntas?

Obrigado!