Machine learning on the edge

With the Jetson Nano Developer Kit

Frank Kelly
HAL24K, PyData Bristol

@norhustla
#PyDataBristol

What is **Edge** computing?

And what's wrong with the Cloud for machine learning?

What is Edge computing?

- “When you generate, collect and analyse data where the [raw] data is generated.”

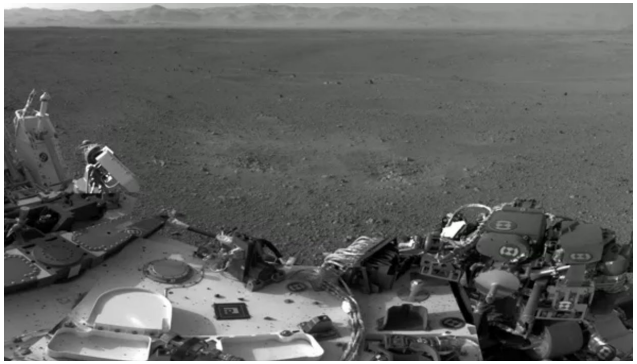
  Bulk of data aggregation and processing at the *edge* of a network

  i.e. not in a data centre

- “Any type of computer program delivering low latency, closer to the requests.”

Typical situations for Edge computing

- Devices with insufficient connectivity
  - Not feasible to connect to cloud
  - High latency
  - Low spectral efficiency*



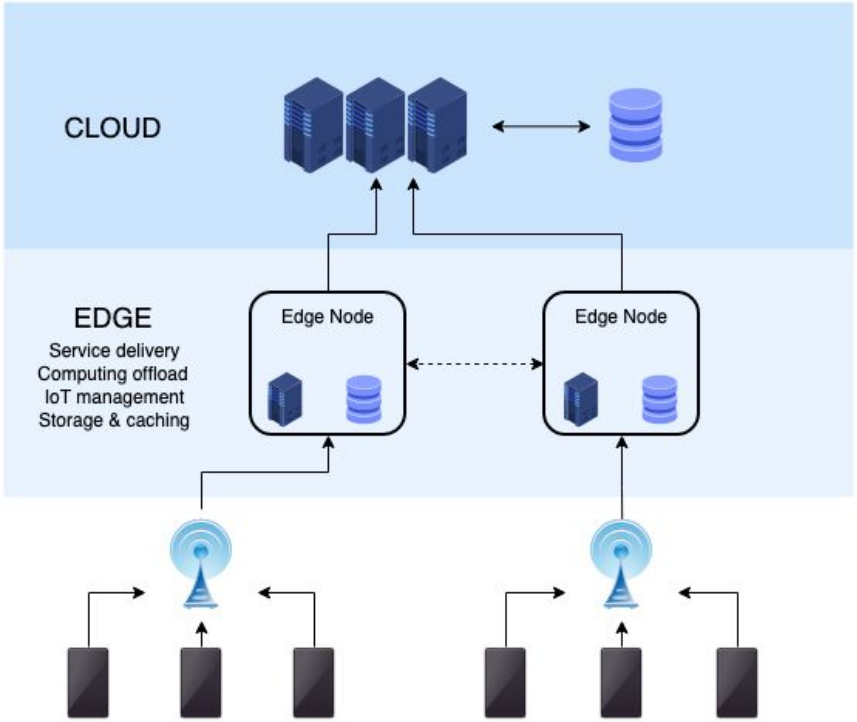
Nasa: Gale crater on Mars taken by Curiosity rover

- Where real-time decisions are needed
- When processing costs in the Cloud are high

* information rate that can be transmitted over a given bandwidth

Industrial uses for Edge computing

- Industrial Internet of Things (IoT)
  - A combination of inter-connected edge devices and cloud computing



wikipedia

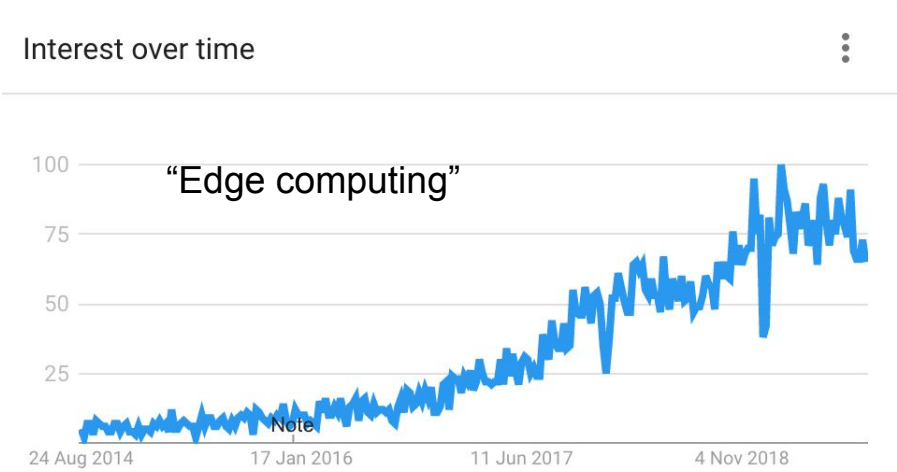
Industrial uses for Edge computing

- Industrial Internet of Things (IoT)
- “Smart factory”
  - Prevent component failure
  - Optimise production
  - Prevent product defects
- “Smart city” / infrastructure
  - Re-route traffic
  - Prevent a leak / flooding event

https://hal24k.com/traffic-solutions

Edge concept on the rise in 2019

- Statista estimates 23 billion Edge connected IoT devices worldwide*
- “Edge is the new Cloud”; predicted annual compound growth rate of 41%*.

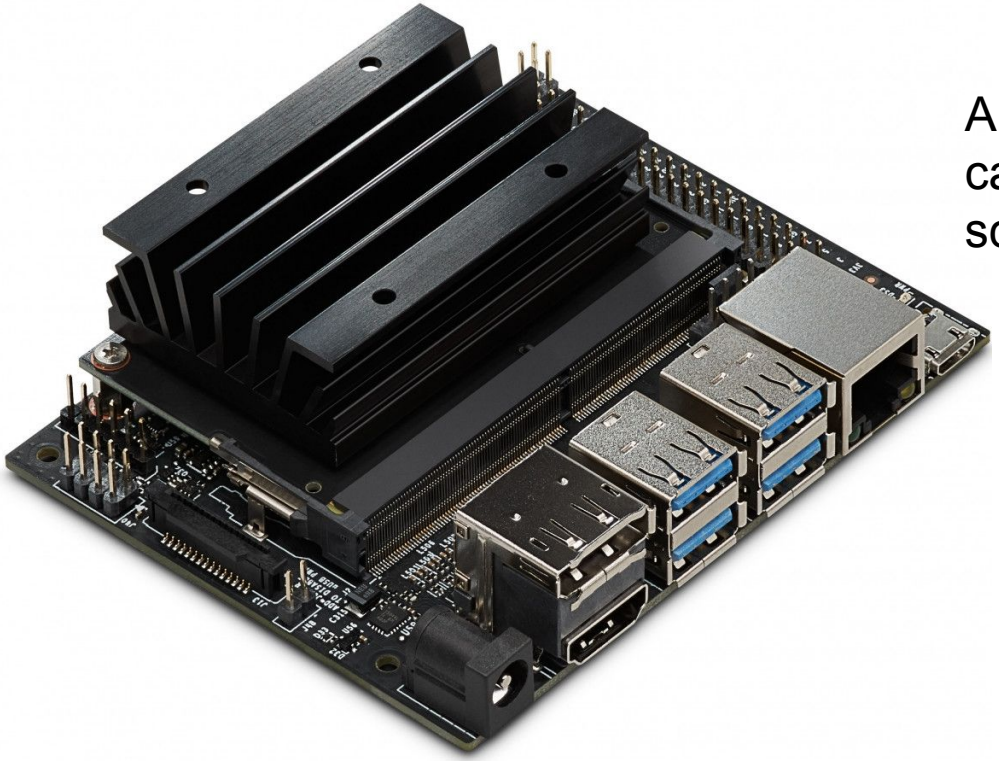


*https://www.forbes.com/sites/forbestechcouncil/2019/01/17/four-predictions-for-edge-computing-in-2019/

Outline of this talk

- Machine learning on the Edge
  - Why should I, and how can I try edge computing?
- How to get started and set up
  - with the NVidia Jetson Nano Developer Kit
- Machine learning applications
  - Home sensor station
- Machine learning on the edge conclusion
  - versus other paradigms

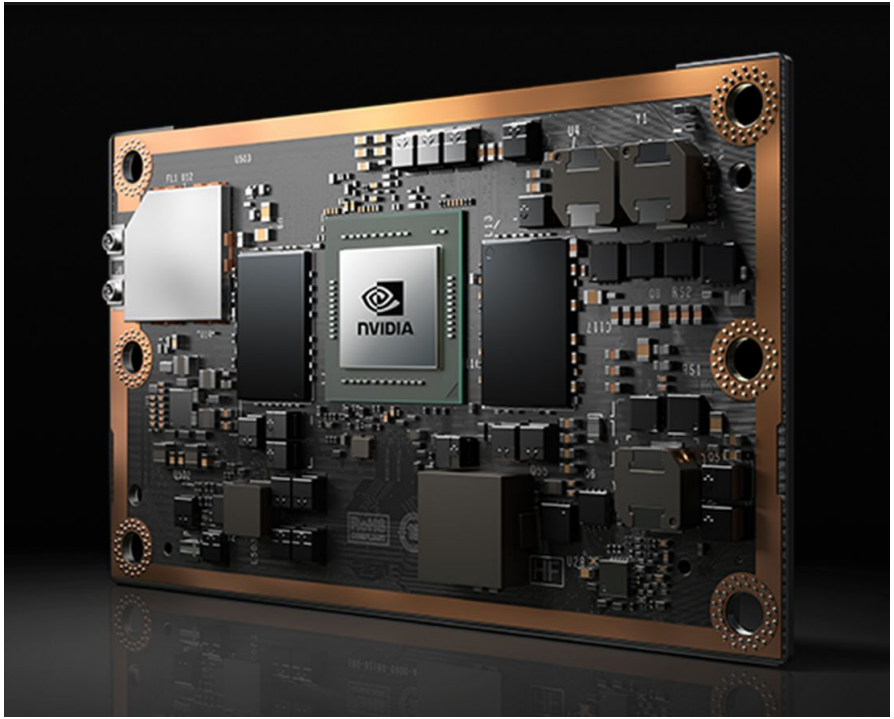
1. Machine Learning ~~on~~ at the Edge?



And why should I care, as a data scientist?

Edge Machine Learning: What is the NVidia Jetson?

- NVidia specialise in computer graphics hardware
- Series of “Jetson” products for commercial projects (e.g. TX2)
- Low-cost, yet very powerful, “AI optimised” compute resources



https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/

Game changer

- Cloud: computationally intensive tasks performed on remote servers
- Edge: data aggregation and simple processing locally only

Now?

- ML, DL and CV* to happen *at the edge*

EDGE COMPUTING

- Basic data visualization
- Basic data analytics and short term data historian features
- Data caching, buffering and streaming
- Data pre-processing, cleansing, filtering and optimization
- Some data aggregation
- Device to Device communications/M2M

CLOUD COMPUTING

- Complex analytics
- Big Data mining
- Sources of business logic
- Machine learning rules
- Advanced visualizations
- Long term data storage/warehousing

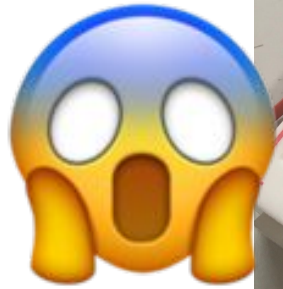
Inference

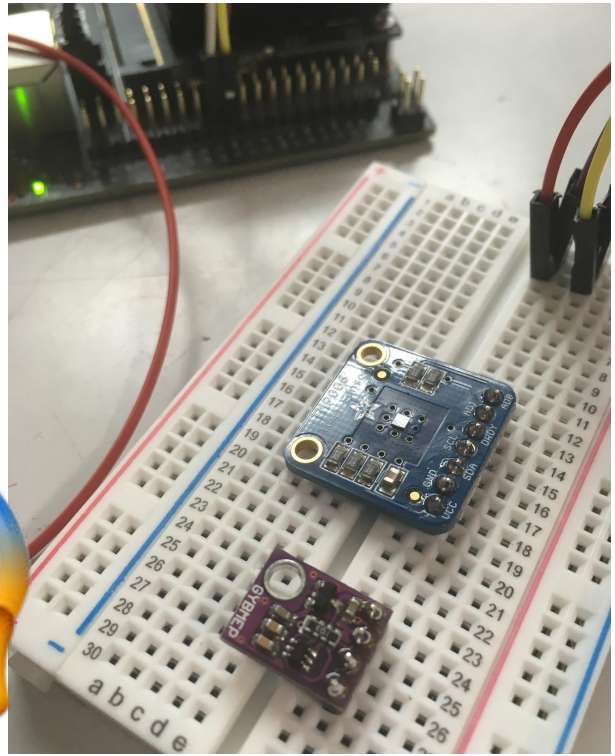
* Machine learning, deep learning and computer vision

Fear of edge machine learning as a data scientist

- Command line skills & dependency installation
- Coding in complicated (non-Pythonic) languages
- Electronics knowledge. Cables. Soldering (!)
- Expensive kit
- Comfortable in my cloud

==> Frustrating experience ?

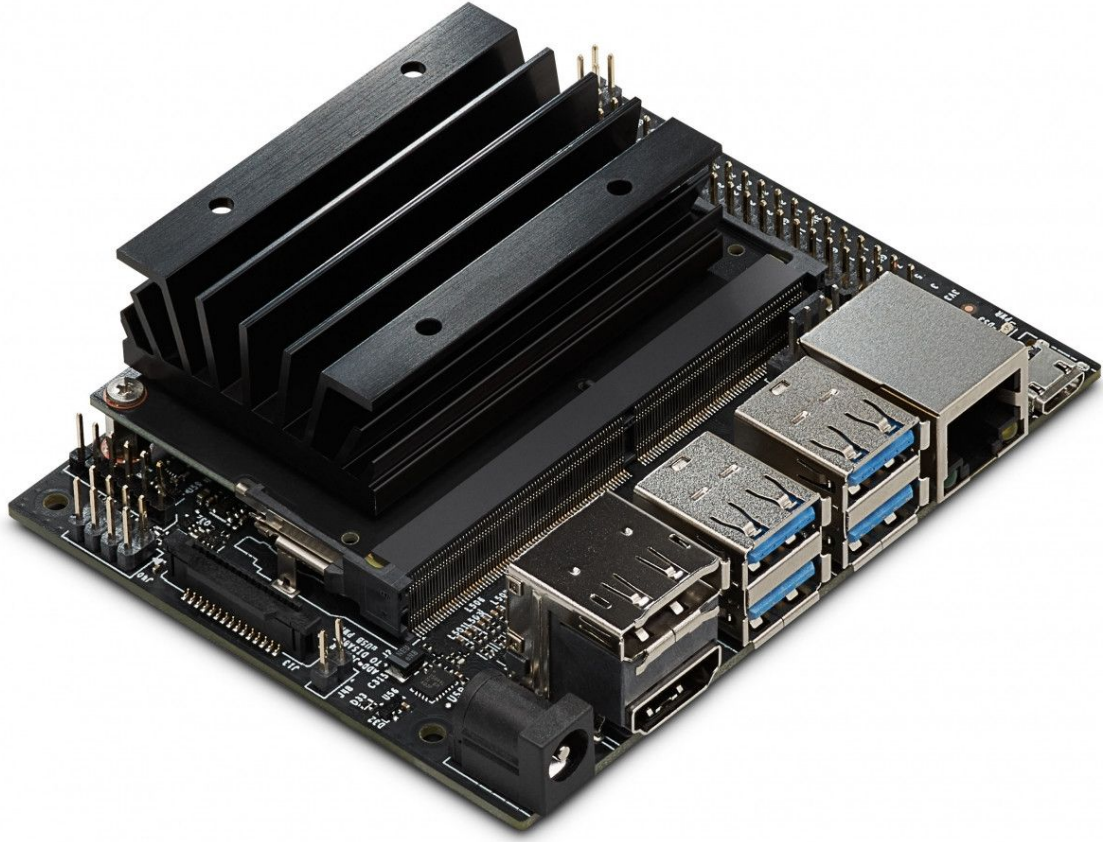




The Jetson Nano

“Democratizing and Disrupting Edge Machine Learning”

- For enthusiasts, hobbyists
- Low cost at $99 (compare $35 for Raspberry Pi)

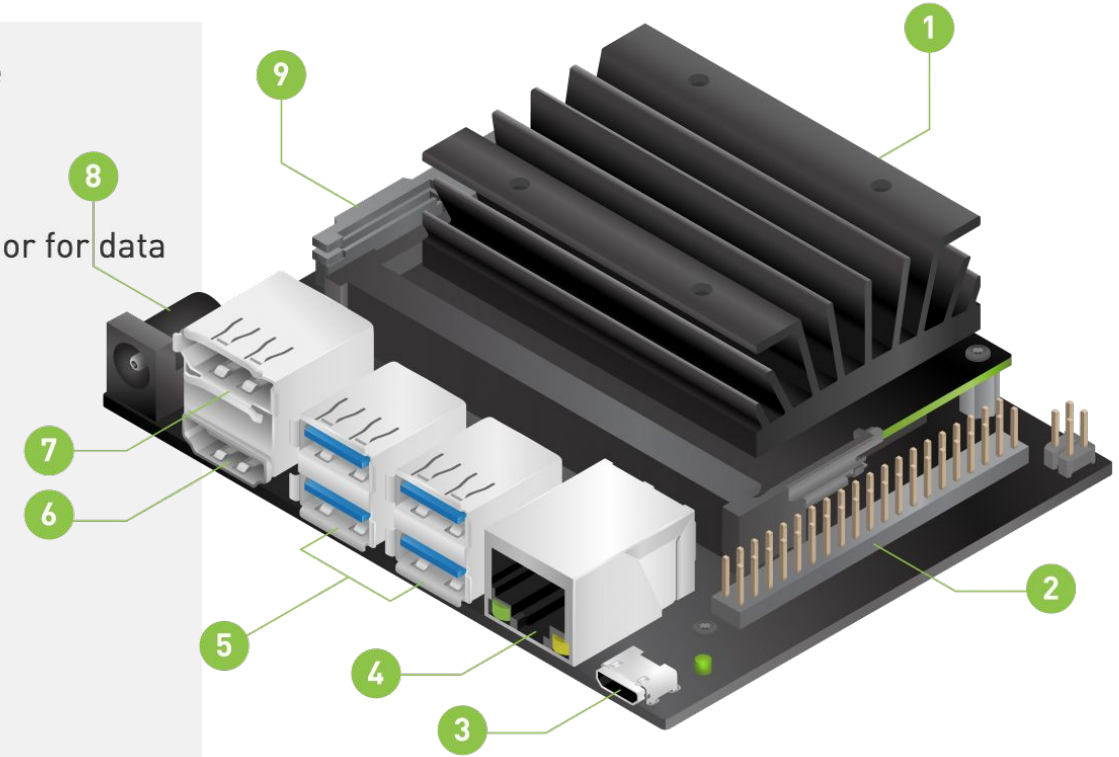


~ a Raspberry Pi with a GPU

https://www.therobotreport.com/nvidias-jetson-nano-democratizing-and-disrupting-edge-machine-learning/

The Jetson Nano Developer Kit

1. microSD card slot for main storage
2. 40-pin expansion header
3. Micro-USB port for 5V power input or for data
4. Gigabit Ethernet port
5. USB 3.0 ports (x4)
6. HDMI output port
7. DisplayPort connector
8. DC Barrel jack for 5V power input
9. MIPI CSI camera connector



+ a 128-core Maxwell GPU

developer.nvidia.com

	Jetson Nano Dev Board	Raspberry Pi 3A+	Raspberry Pi 3B+
AI Performance	472 GFLOPS	21.5 GFLOPs (est*)	21.4 GFLOPs (est*)
CPU	1.4 GHz 64-bit Quad-Core ARM Cortex-A57 MPCore	1.4 GHz 64-bit Quad-core ARM Cortex-A53	1.4 GHz 64-bit quad-core ARM Cortex-A53
GPU	128-Core Nvidia Maxwell	Broadcom VideoCore IV	Broadcom VideoCore IV
RAM	4GB LPDDR4	512MB LPDDR2 SDRAM	1GB LPDDR2 SDRAM
GPIO Header	40-pin	40-pin	40-pin
Board Dimensions	100 X 79mm	65 X 56mm	85 x 56mm
Wireless	None	Dual-band 802.11ac wireless LAN, Bluetooth 4.2/BLE	Dual-band 802.11ac wireless LAN, Bluetooth 4.2

https://www.maketecheasier.com/nvidia-jetson-nano-vs-raspberry-pi/

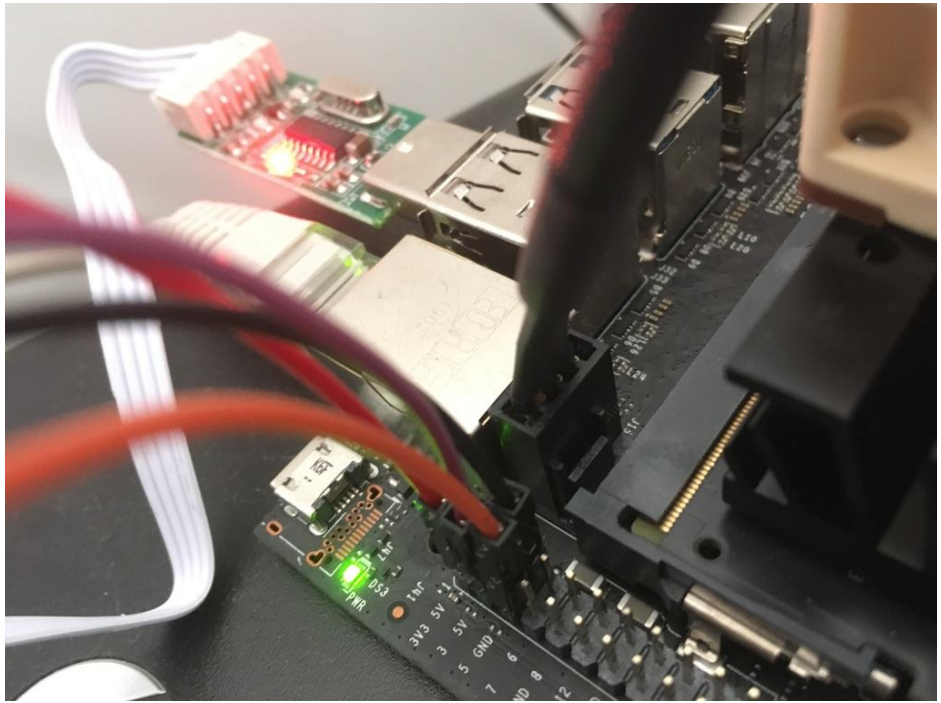
What is your goal with the Nano?

- The real intention is for machine learning
  - Think about applications
- Remember inference only possible, not model training
- Keep it simple (ish)

What is your goal with the Nano?

- Take an existing Python-based machine learning application,
  - Run it on the Jetson Nano
  - Minimal modifications required
  - Test out the performance
- Build your own home ML inference platform
  - Home sensing station with predictive analytics

2. Jetson Nano: How to get started and set up



How to get started

- What you get and hardware setup - out of the box
- Additional components -

  what do you need and how much does it cost?

- Software setup -

  Deep learning frameworks and dependencies

Hardware set up: out of the box

Minimum requirements:

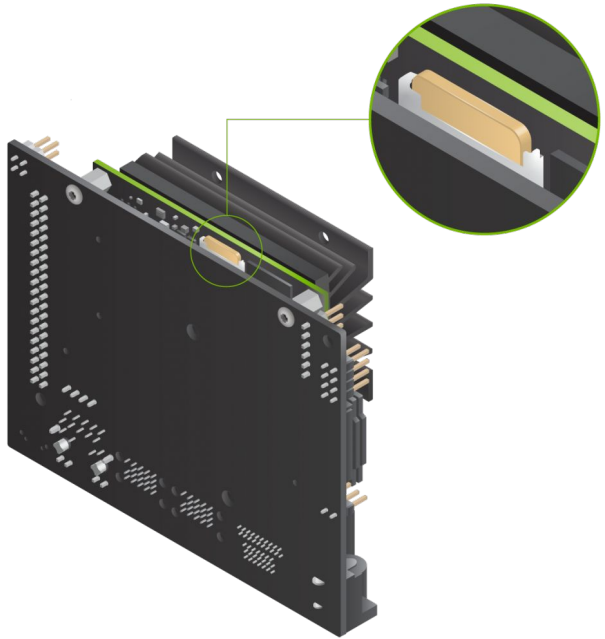
- Ethernet cable
- 5V 2.5A micro-USB power cable
- 16GB micro-SD card

First step:

- Download and flash MicroSD
  - with NVidia image

Starting up for the first time

- Insert the micro-SD card
  - Located under the heatsink
- Connect the micro-USB power supply
- Ubuntu boots up
  - First time config



Hardware set up: Additional components

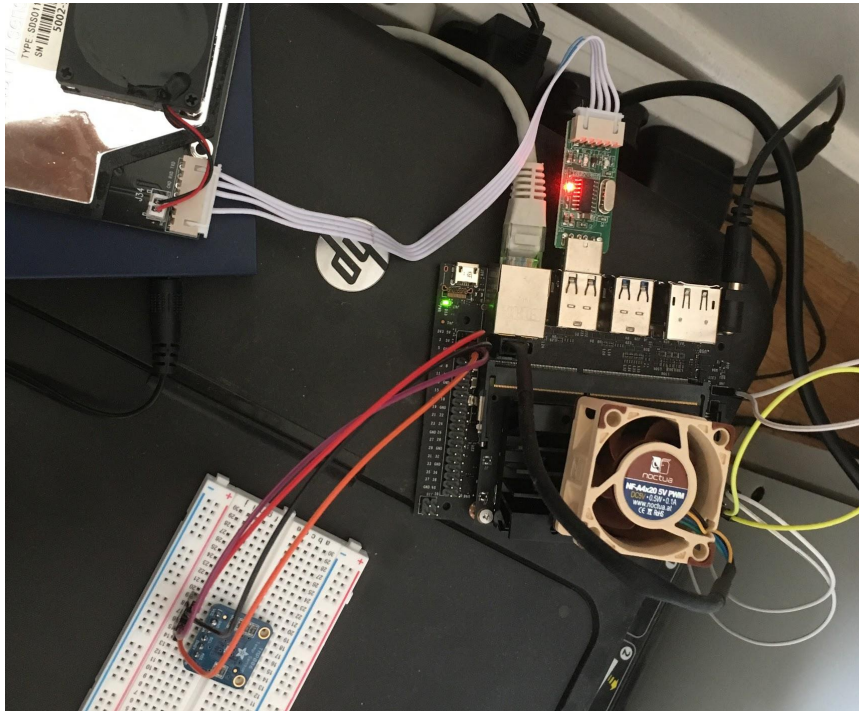
Avoid auto shut-off / overheating:

- Barrel 5V power supply (£15)
- A cooling fan (£12)

Connect various sensors:

- Adafruit, etc (£…), cables (£3)
- Raspberry Pi CSI-2 camera (£18)

Disk space: MicroSD 64GB



Software set up: System initialisation

- Install system packages and prerequisites
  - e.g. Git, HDF5

- Configure your Python environment
  - e.g. Virtual environments with virtualenv and virtualenvwrapper

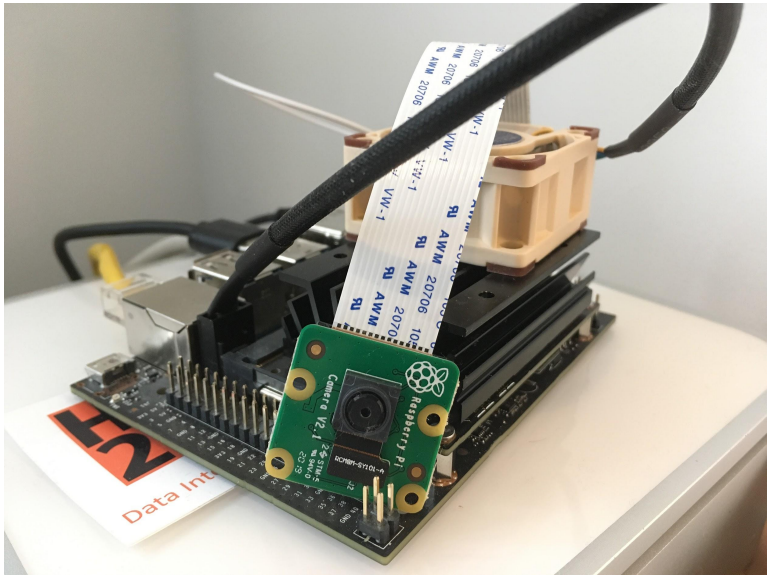
- Move on to data science and machine learning libraries

Software set up: Deep learning libraries

- The same CUDA libraries used for acceleration by Python-based deep learning frameworks are supported on the Jetson Nano

ML applications: Jetson Inference

- Try out the camera
  - Flip method if upside down
  - CSI-2 or USB webcam



- Object recognition camera application
  - Uses Google ImageNet to recognise objects
- Pedestrian detection camera application
  - Picks up humans in frame

http://www.image-net.org/about-overview
https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks

Software set up: Deep learning libraries

- First install Numpy
  - Currently no pre-built versions of Numpy for the jetson Nano

```
Getting started with the NVIDIA Jetson Nano                    Shell
1 $ pip install numpy
```

- For tensorflow; do NOT try **$pip install tensor flow-gpu**

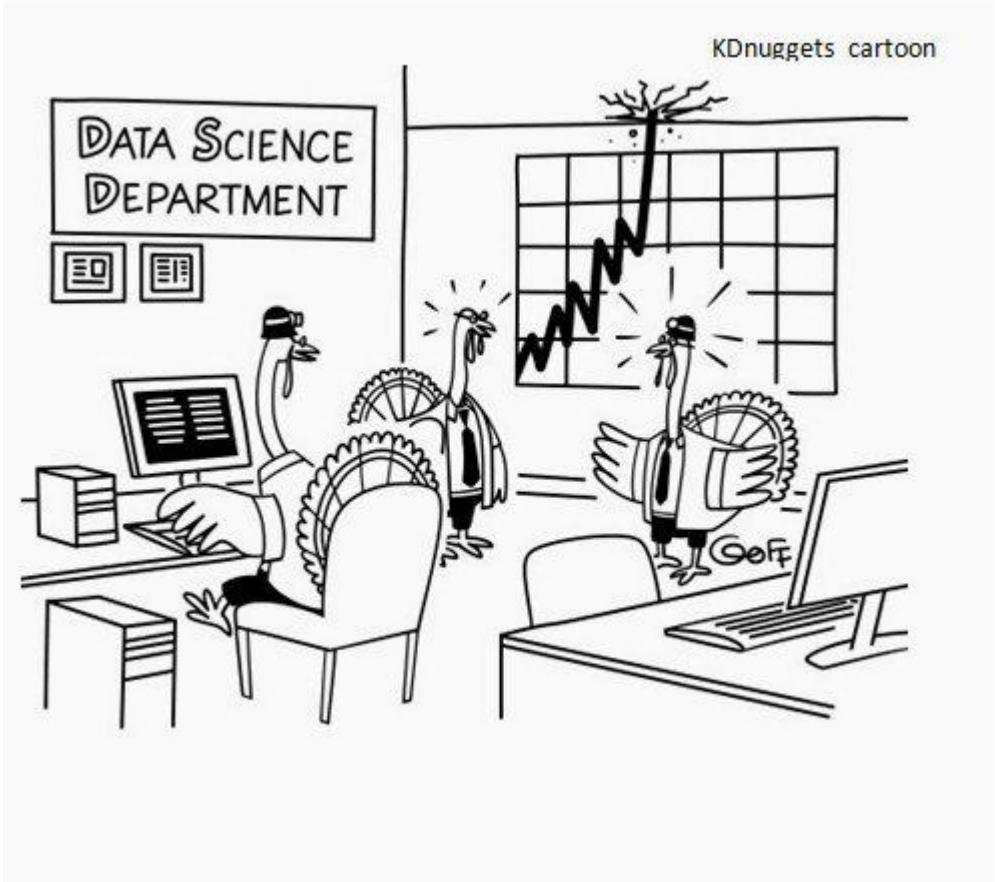
```
Getting started with the NVIDIA Jetson Nano                    Shell
1 $ pip install --extra-index-url https://developer.download.nvidia.com/compute/redi
  st/jp/v42 tensorflow-gpu==1.13.1+nv19.3
```

- Install SciPy & Keras

```
Getting started with the NVIDIA Jetson Nano
1 $ pip install scipy
2 $ pip install keras
```


https://www.pyimagesearch.com/

3. A machine learning project on the Jetson Nano Developer kit



Machine learning project on the Jetson Nano

- Air quality predictions
  - Can I predict the air quality
  - Super local
  - Over the next hour or two
  - Both inside and outside my home

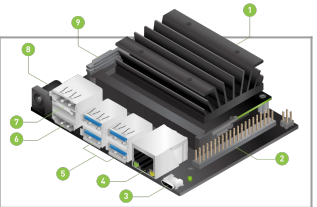
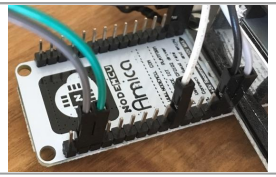
- First, I need data! (cloud free)
  - Requires sensors, nodes (?) and data storage

The burden of particulate air pollution in the UK in 2008 was estimated to be equivalent to nearly 29,000 deaths ....

It has been estimated that removing all fine particulate air pollution would have a bigger impact on life expectancy in England and Wales than eliminating passive smoking or road traffic accidents.

https://laqm.defra.gov.uk/public-health/public-health-impacts.html

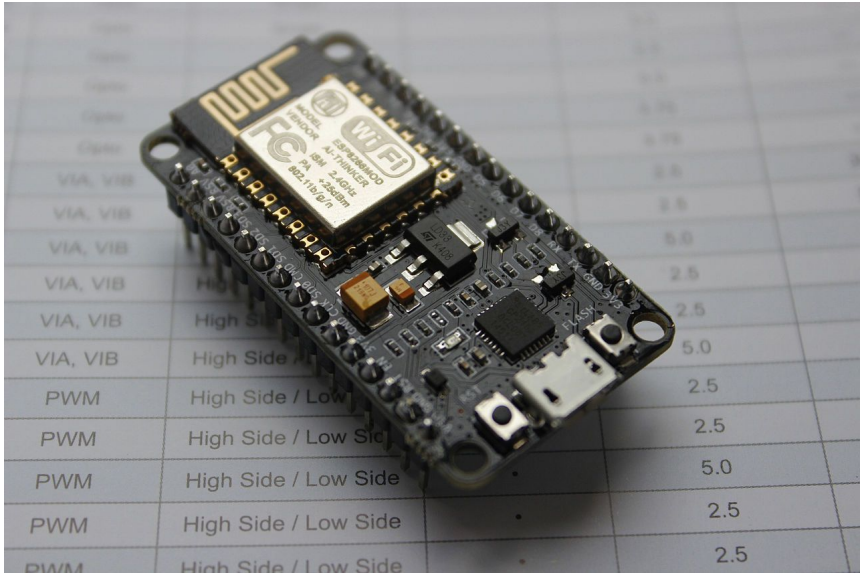
An additional paradigm

Edge computing	Data is processed on the device or sensor itself without being transferred anywhere. Physically close to the sensors (e.g. video feed).	
Cloud computing	The use of a “collective pool” or network of remote servers hosted on the Internet to store, manage and process data rather than a local server. Minimal management effort.	
Fog computing	Data processing from the edge to the cloud. Introduces a “fog node” or “IoT gateway”, that processes the data and is situated within the LAN.	

Data collection: use of IoT nodes with sensors

- Connect a NodeMCU
  - An open-source Internet of Things (IoT) platform
  - Uses Lua scripting language

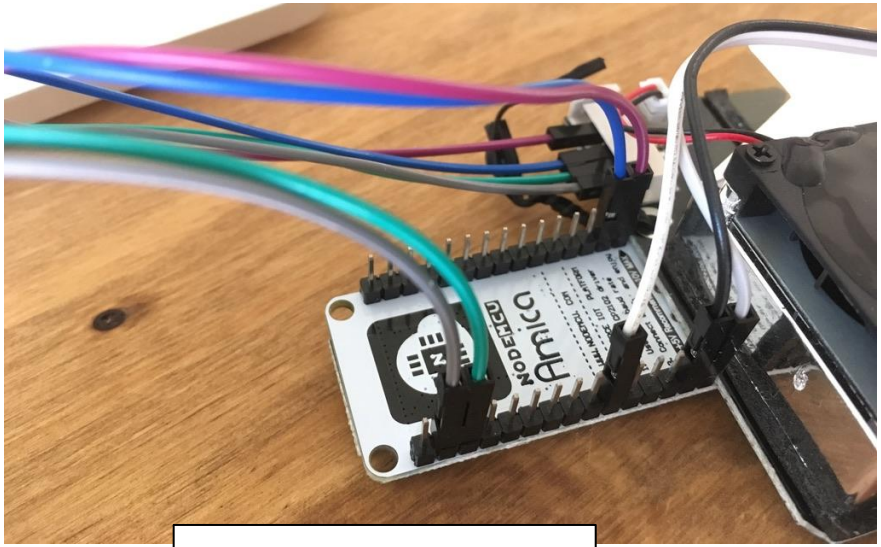
- Fog computing concept
  - “Similar functionality as the Cloud, but physically closer to Things”



Data collection: Set up NodeMCU

- Adjust Lua config files
  - Initialisation
  - Per sensor

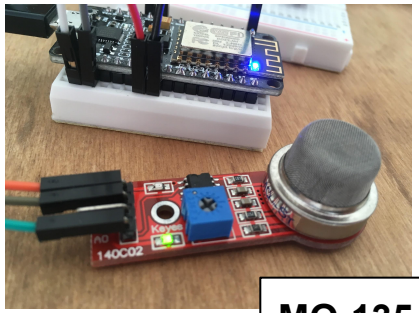
- Flash the device
  - Test
  - And repeat...



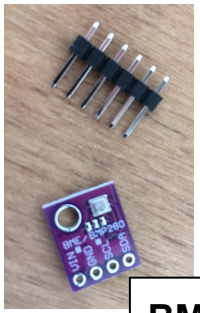
NodeMCU ESP8266

Data collection: Sensors

- Choose relevant sensors
  - Soldering is relatively painless

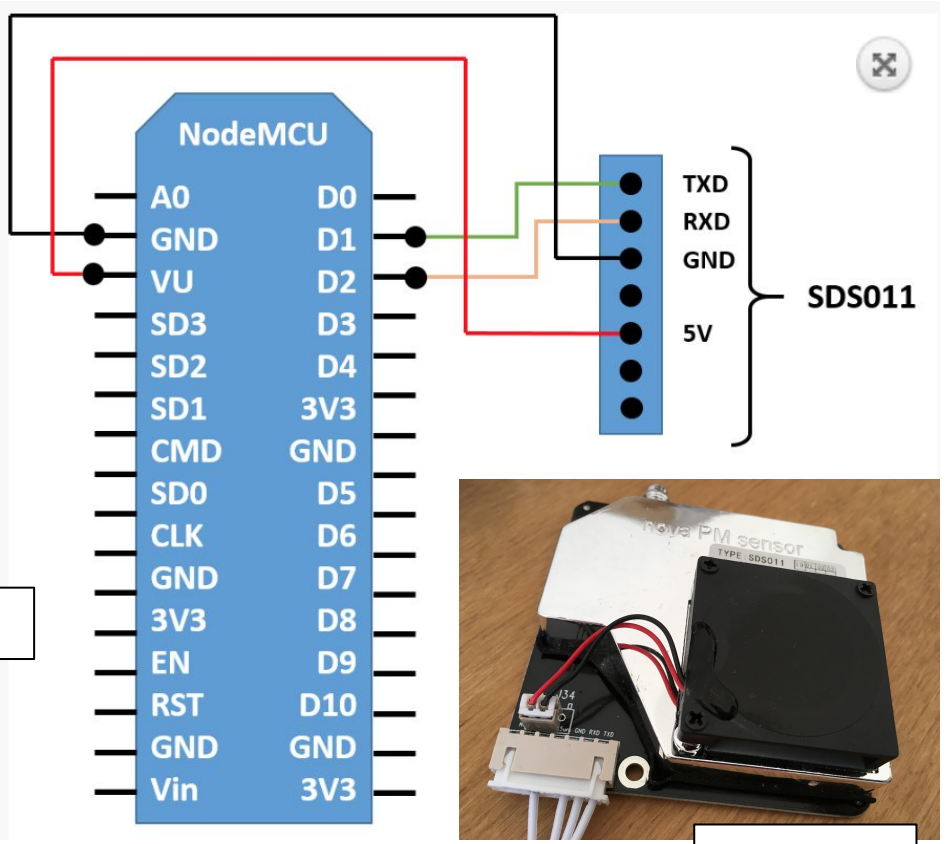


MQ-135



BME-280

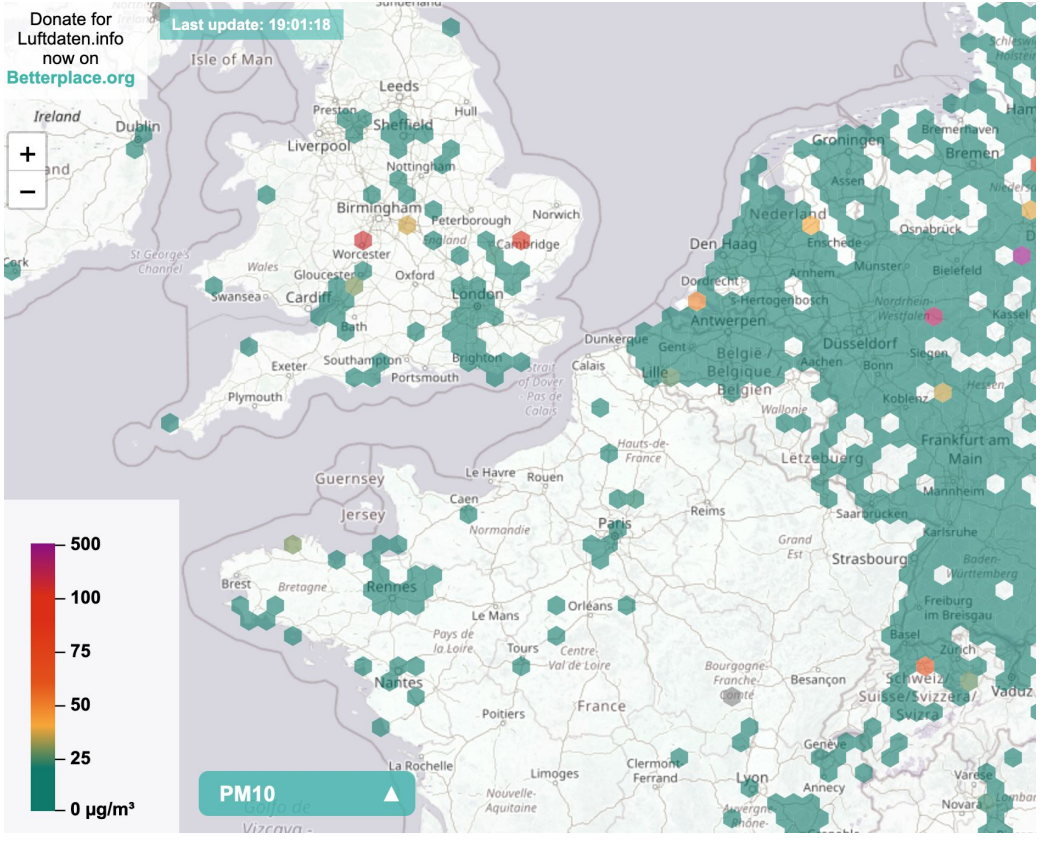
- Connect to NodeMCU, or Jetson Nano directly
  - Ensure correct pins connected



SDS-011

An aside on citizen science

- Citizen science via IoT and the cloud
- Set up your own air quality sensor at luftdaten.info



@norhustla
#PyDataBristol

Data collection: Set up a database on the Jetson

- InfluxDB - distributed, time series database. Queries are SQL-like.
- Either poll to get data from the Jetson Nano or push data directly to your database

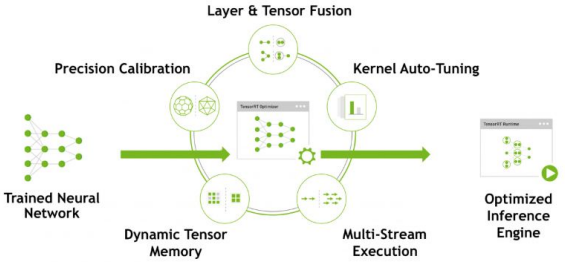
```
Sep 08 19:11:10 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:11:09', 'fields': {'T': 22.57, 'P': 1021.031, 'QNH': 1022.6, 'H': 46
(node_mcu_sensor_stuff) frank@frank-jetson:/opt/poll-bme280$ sudo systemctl status poll-bme280.service
● poll-bme280.service - Poll Service
   Loaded: loaded (/opt/poll-bme280/poll-bme280.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2019-09-08 19:11:08 BST; 5min ago
 Main PID: 29529 (poll-bme280.sh)
    Tasks: 2 (limit: 4190)
   CGroup: /system.slice/poll-bme280.service
           ├─29529 /bin/bash /opt/poll-bme280/poll-bme280.sh
           └─29550 python poll.py

Sep 08 19:15:07 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:07', 'fields': {'T': 22.51, 'P': 1020.988, 'QNH': 1021.957, 'H':
Sep 08 19:15:18 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:18', 'fields': {'T': 22.51, 'P': 1021.006, 'QNH': 1021.975, 'H':
Sep 08 19:15:28 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:28', 'fields': {'T': 22.52, 'P': 1020.991, 'QNH': 1021.96, 'H': 4
Sep 08 19:15:38 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:38', 'fields': {'T': 22.51, 'P': 1020.998, 'QNH': 1021.967, 'H':
Sep 08 19:15:49 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:49', 'fields': {'T': 22.51, 'P': 1020.986, 'QNH': 1021.955, 'H':
Sep 08 19:15:59 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:15:59', 'fields': {'T': 22.51, 'P': 1020.987, 'QNH': 1021.956, 'H':
Sep 08 19:16:10 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:16:10', 'fields': {'T': 22.51, 'P': 1020.985, 'QNH': 1021.954, 'H':
Sep 08 19:16:20 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:16:20', 'fields': {'T': 22.5, 'P': 1020.995, 'QNH': 1021.964, 'H': 4
Sep 08 19:16:31 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:16:30', 'fields': {'T': 22.5, 'P': 1021.016, 'QNH': 1021.985, 'H': 4
Sep 08 19:16:41 frank-jetson poll-bme280.sh[29529]: {'measurement': 'bme280', 'time': '2019-09-08T18:16:41', 'fields': {'T': 22.49, 'P': 1020.994, 'QNH': 1021.963, 'H':
lines 1-19/19 (END)
```

Model training (cloud) and deployment (edge)

- Make use of PaperSpace or similar cloud-based machine learning service
  - Create an NVidia VM
  - Load this into a Docker container
  - Spin up a high spec, multi GPU instance
  - Launch model training; e.g. auto-encoder LSTM; any deep learning framework
  - Keep an eye on the clock $0.50 / hour (example)
  - Output an NVidia TensorRT model
- Send the model to your Jetson Nano
  - Run and perform inference using fresh data

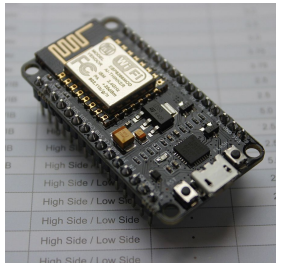
Paperspace

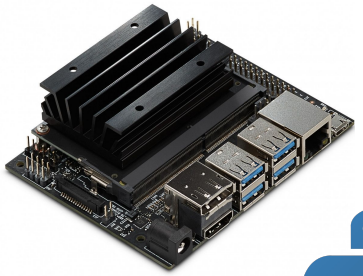


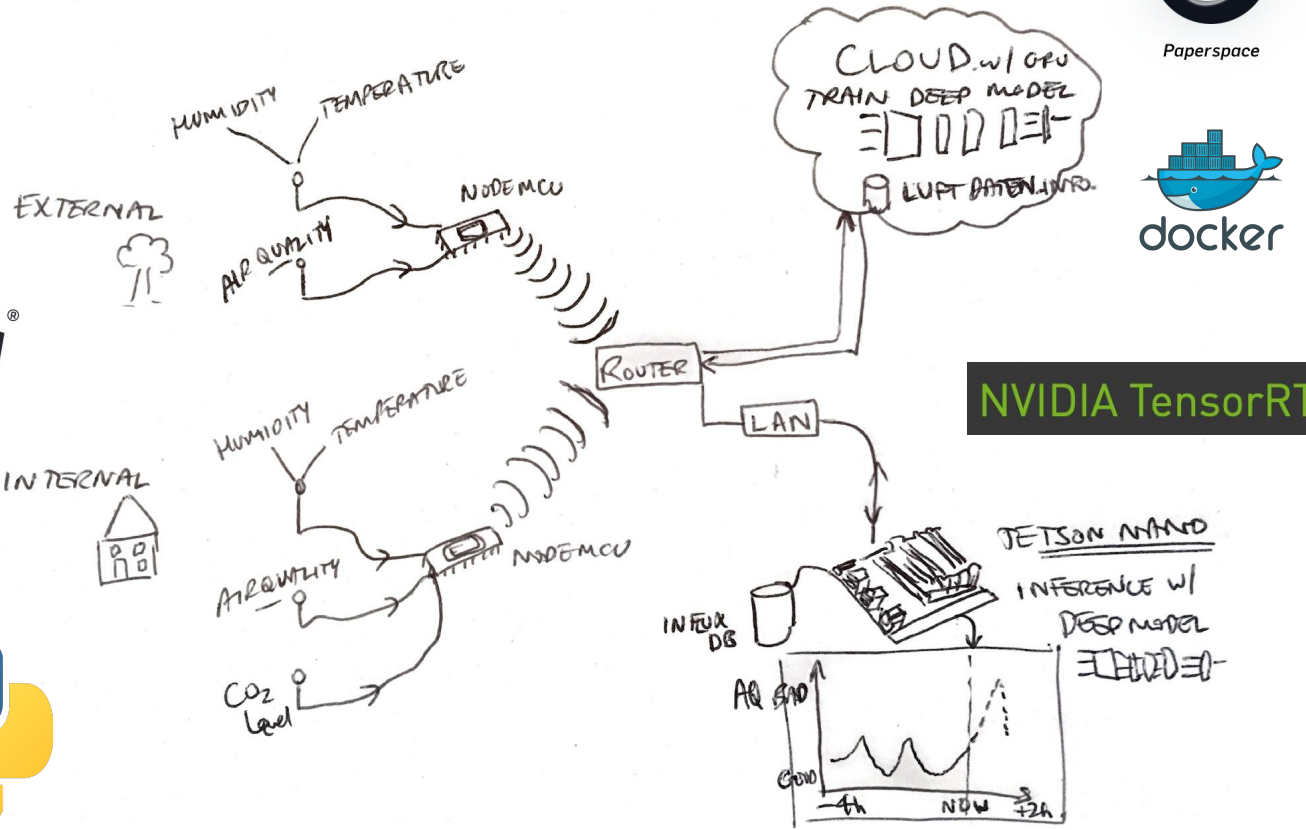
(Click to Zoom)

https://developer.nvidia.com/tensorrt

Fog-based ML pipeline complete!







Paperspace

docker

influxdata®

NVIDIA TensorRT

@norhustla
#PyDataBristol

ML applications: Transfer learning and FastAI

- Transfer learning
  - Load in, pre-format your data

    Or, capture some for a while, then pre-format

  - Load in a (pre-trained) model
- Perform inference, at the edge!
- Store and process results,
- Insight and display to the world
  - Grafana or similar

4. Considering Edge vs other options

For your machine learning project

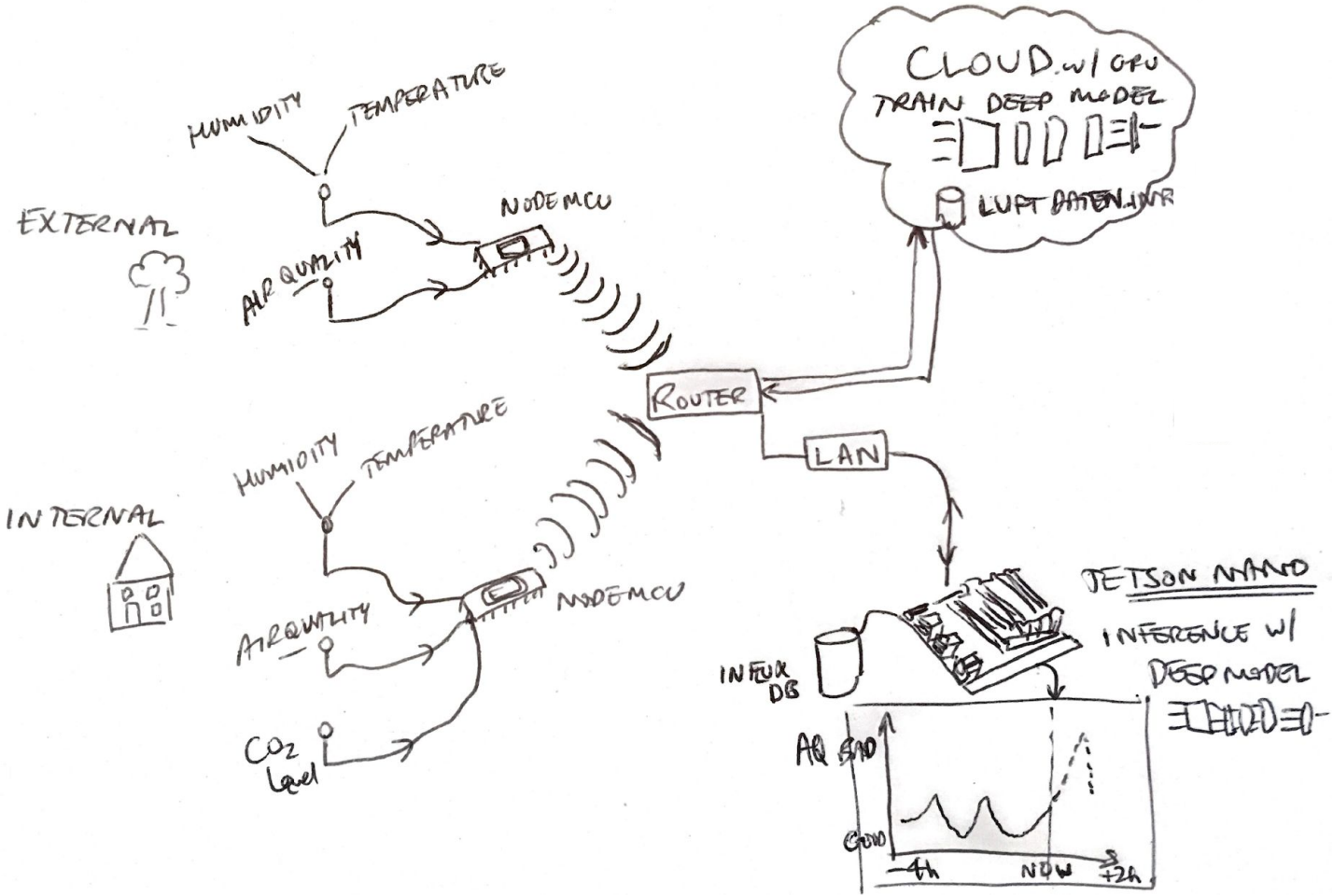
Think: what are you trying to do?

You have four main options for your machine learning project

Cloud computing + No theoretical GPU processing power limit + Excellent for model training on huge, varied datasets - Latency (comms) - Hosting and compute costs	**On site computing (server)** + Lower running costs - Limited by the machine's power - High one-off cost (e.g. £2K)
Edge computing + No latency issues + GPU processing for inference + Low cost - Slow for training	**Fog computing** + Share tasks between edge and cloud + Medium-low cost + Low latency - Complexity of system & management thereof

Edge computing does not replace cloud computing

- Instead, an ideal scenario is foggy:
  1. Real world data collected from local sensors, sent to node units
  2. NodeMCU transmits to both local and cloud databases
  3. Machine learning models periodically trained on the historical data in a cloud environment (high spec for short period)
  4. Periodically push out models to an edge device
  5. Edge device uses inference on recent sensor data to generate immediate insight.



Summary

- Edge computing is a powerful concept
  - Expect to see more of it with machine learning applications
  - GPU inference is now possible at the edge
- As a data scientist, you should try the Jetson Nano Developer kit
  - Low cost and “easy” way to get into the world of Edge computing
- Jump into machine learning on the edge
  - Plenty of resources (thanks, Raspberry Pi community)

References

https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

Setting up the Jetson Nano: https://www.jetsonhacks.com/ (includes YouTube)

Setting up for deep learning and computer vision: https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/ (his book to come out soon)

Scripts to configure Nano for FastAI: https://github.com/brtknr/fastai-jetson-nano .

Citizen Air quality sensor: https://luftdaten.info/en/home-en/