AI/ML introduction

Athula Wickremasinghe

02/25/2021

Based on Oracle (https://blogs.oracle.com/bigdata/difference-ai-machine-learning-deep-learning)

Brief Introduction to ML

ML

Supervised Learning

Learning with a known output

Regression
- Forecasting
- Predicting
- New Insights
- Process Optimizing
- ....

Classification
- Categorizing
- Labeling
- Seperations
- ....

Semi supervised Learning

Learning from large data which has small set of data with known output

Reinforcement Learning

Self learning with some supervision
- Agent Learns itself by getting rewards
-
- ....

Unsupervised Learning

Learning without a known output

Clustering
- Finding hidden patterns
- Identifying data structure
- Grouping data
- ....

Dimensionality reduction
- Reduce dimensional space from a high dimensional space
- Reduce number of features to make it easy

supervised learning

Input data

Annotations

These are apples

Model

Prediction

Its an apple!

unsupervised learning

Input data

Model

environment

agent

actions

rewards
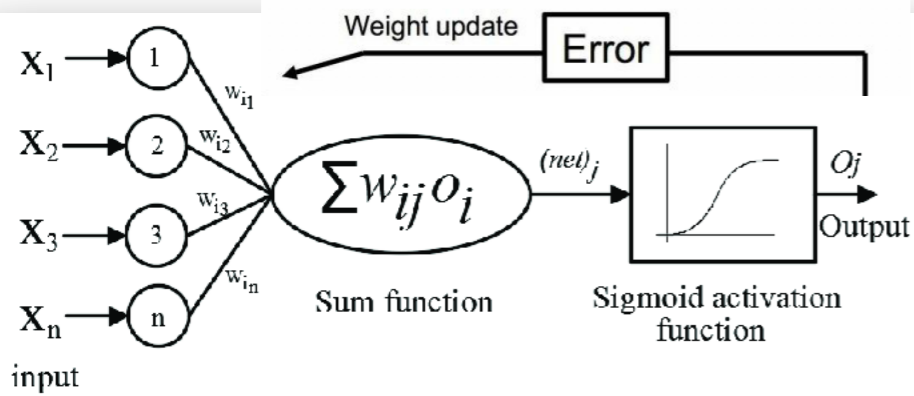
observations

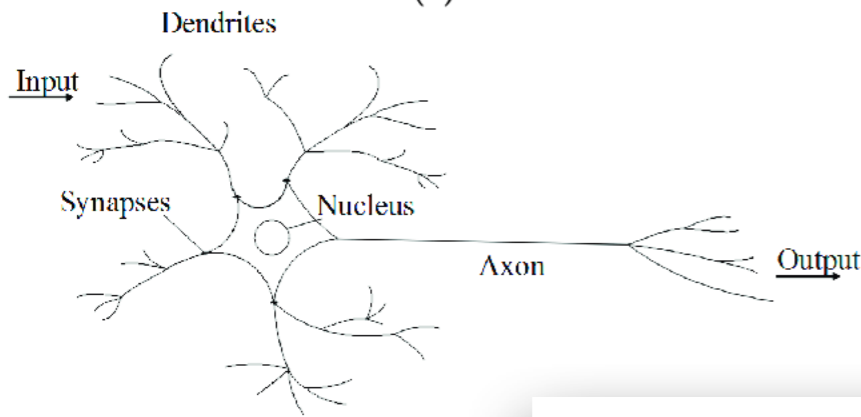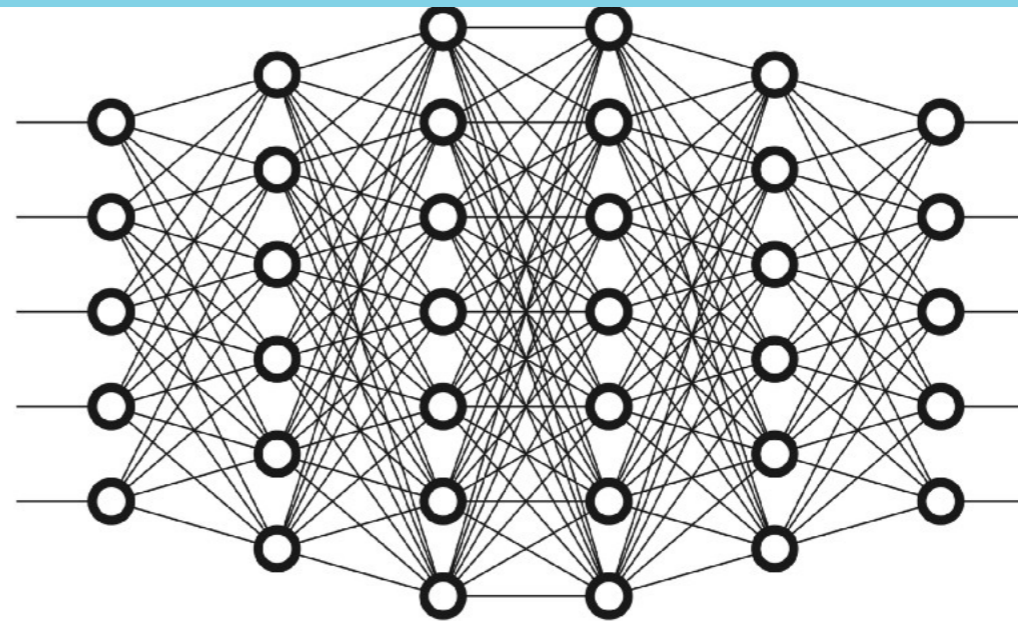🟦 **Fermilab**
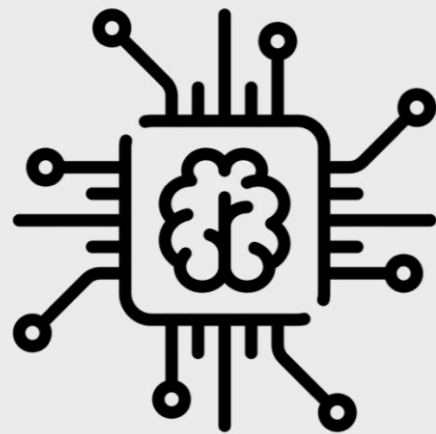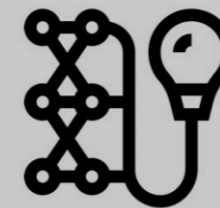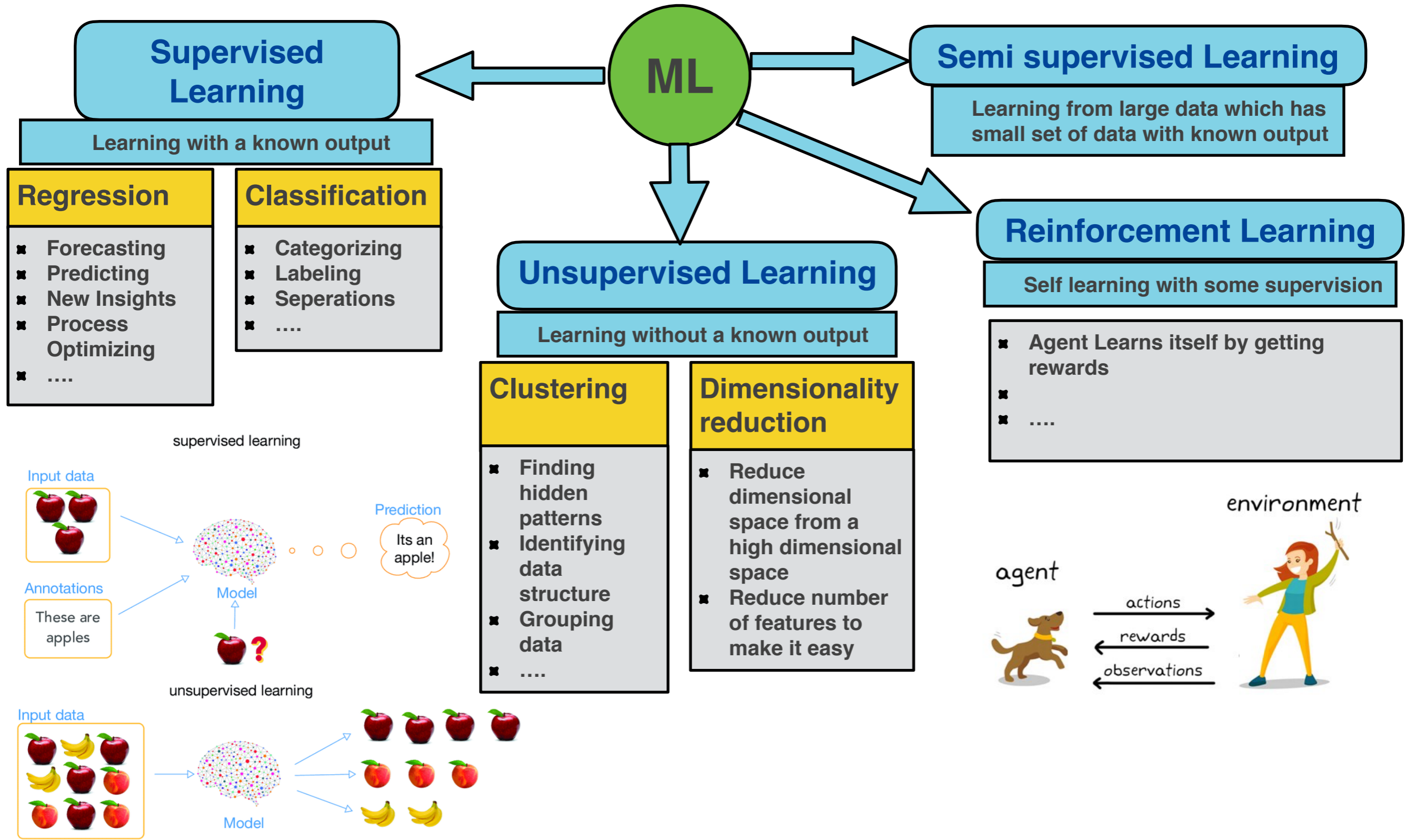
Brief Introduction to Neural Network

(a)

(b)

Input

Output

Hidden Layers

$$w_{new} = w_{old} + \eta \cdot \nabla Error$$

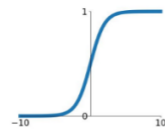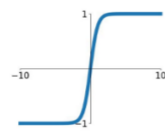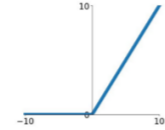Weights are updated according to the backpropagation algorithm
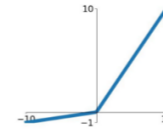
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
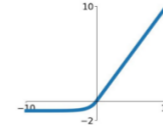
tanh
$\tanh(x)$

ReLU
$\max(0, x)$

Leaky ReLU
$\max(0.1x, x)$

Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

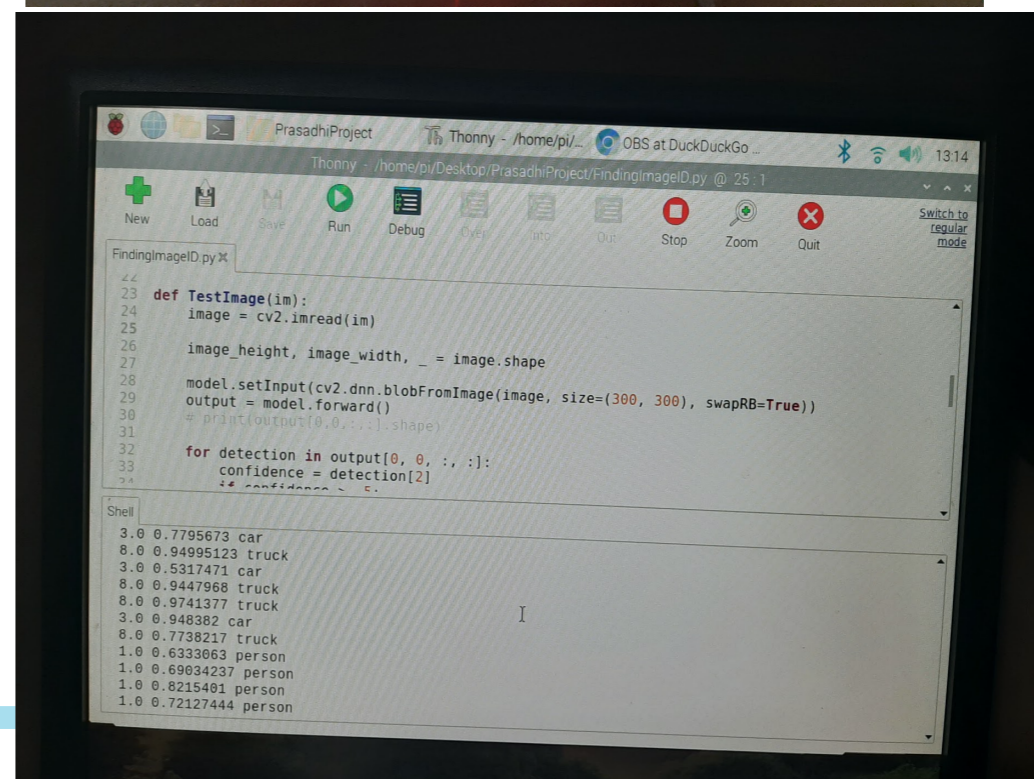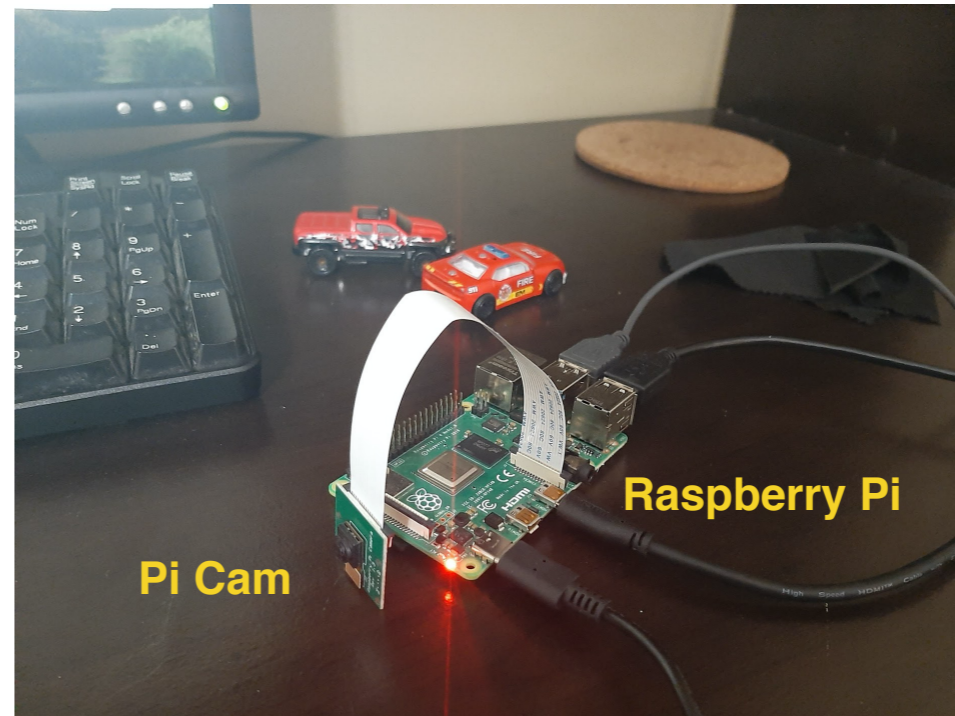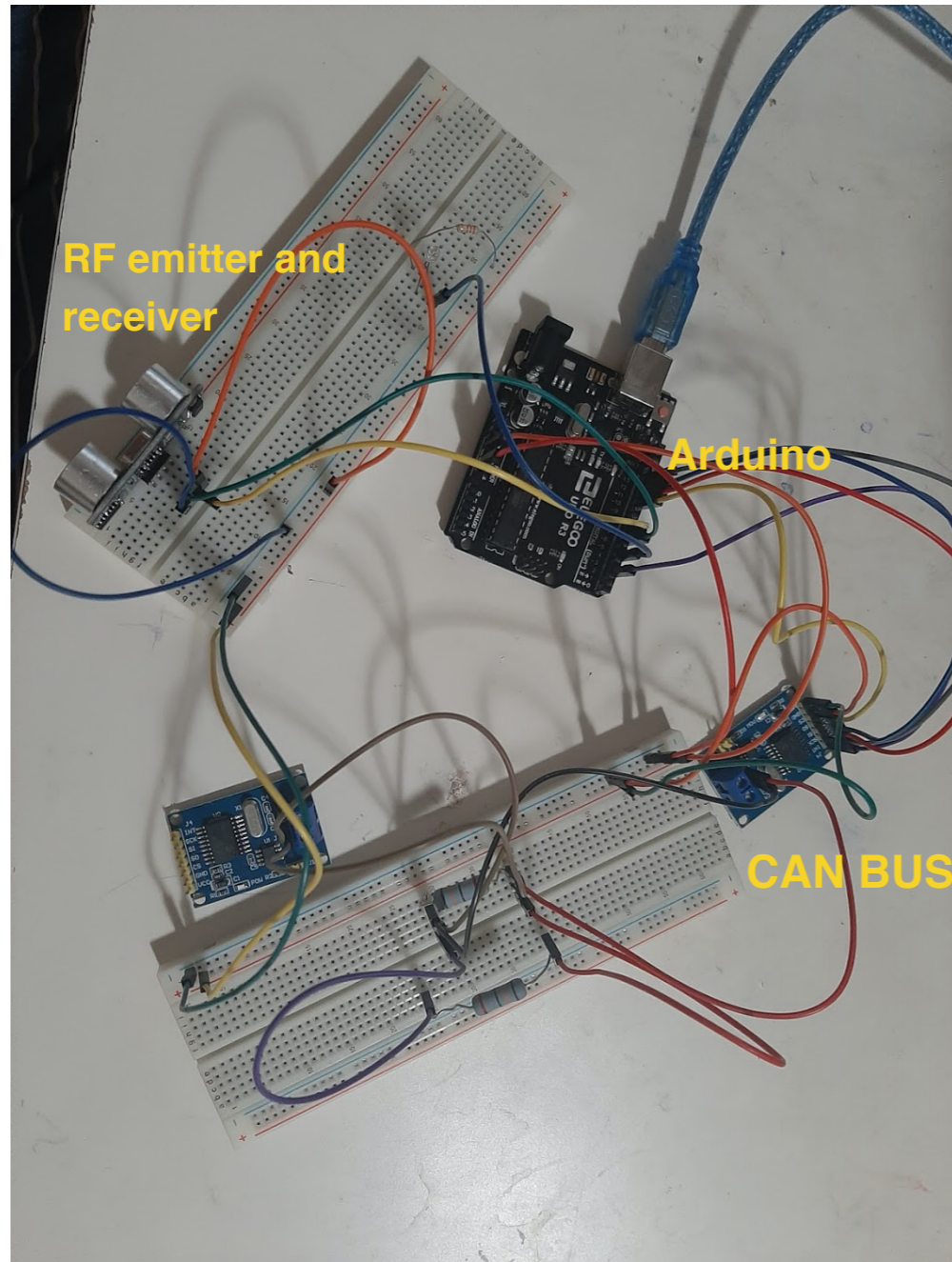Activation function

Example of an activation function

$$\frac{1}{1 + \exp\left(-\sum_j w_j x_j - b\right)}.$$

Network tuning

» **Learning rate**

» **Number of nodes**

» **Number of hidden layers**

» **Bias**

» **Batch size**

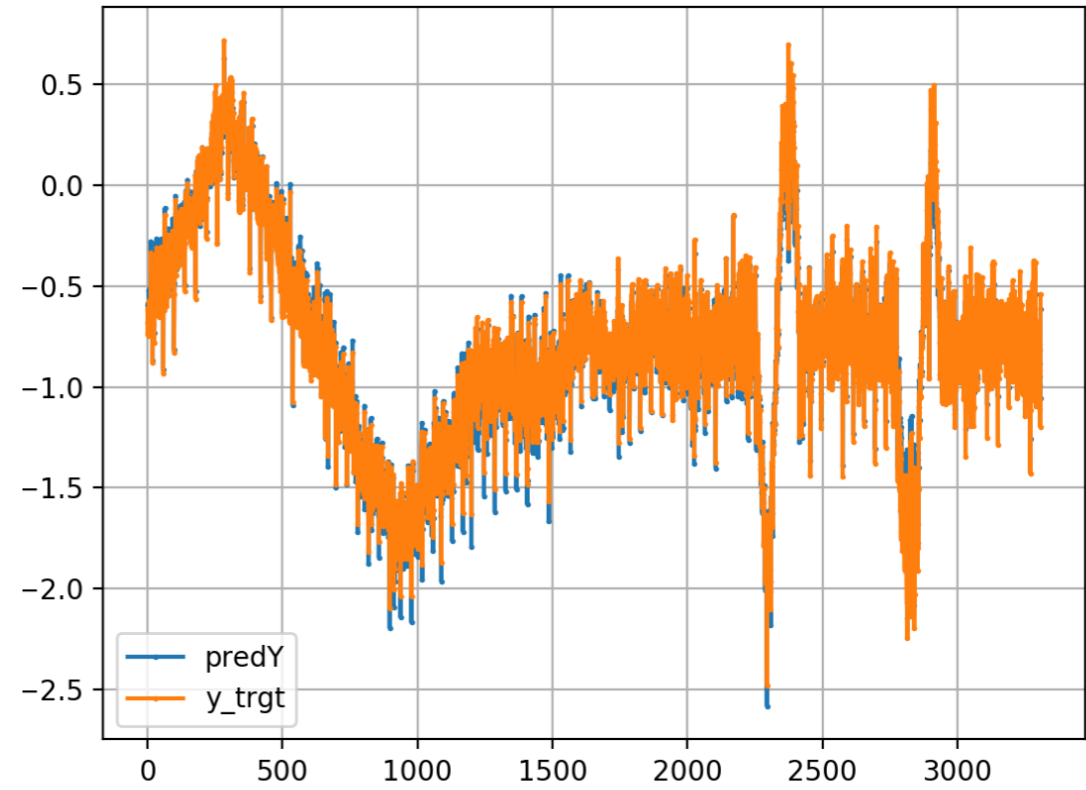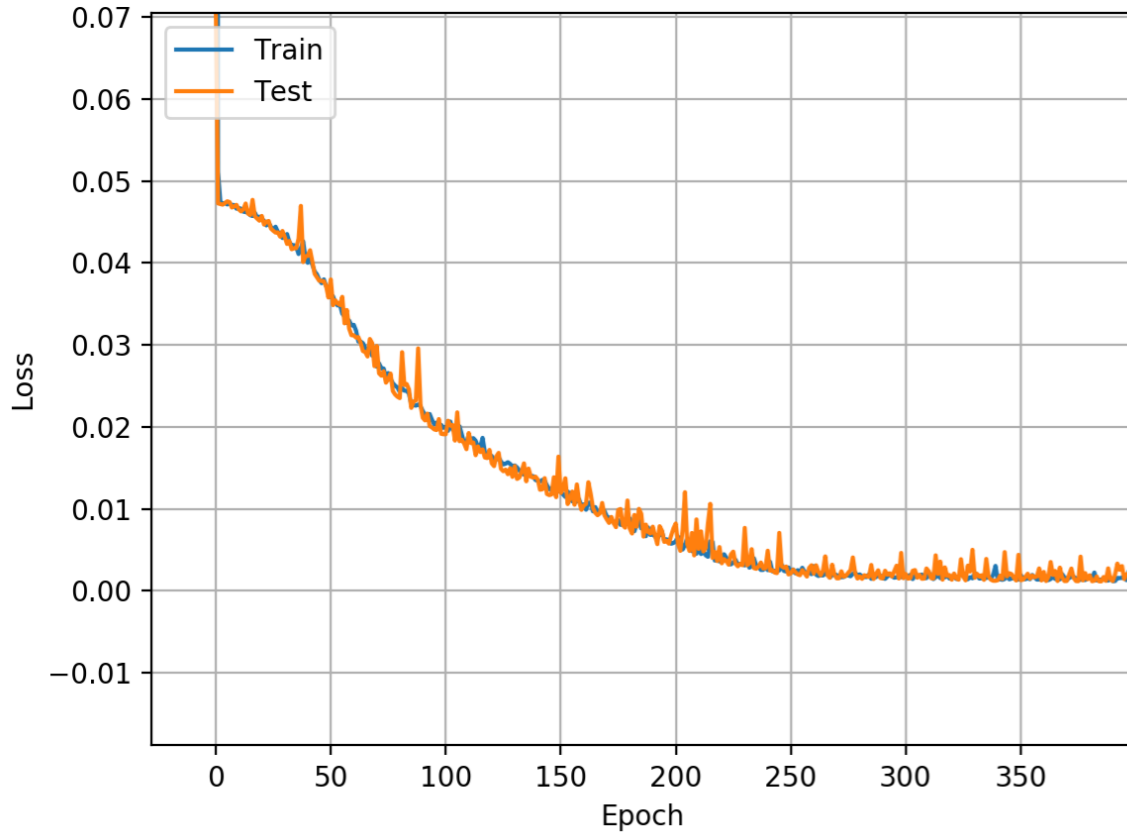🔷 **Fermilab**

My IoT project for fun

This is a small project of showing how to take an action based on identifying an object and the distance to that object

CNN Model: MobilNet_SSD:v2_coco_2018

RF emitter and receiver

Arduino

CAN BUS

Raspberry Pi

Pi Cam


```
23  def TestImage(im):
24      image = cv2.imread(im)
25
26      image_height, image_width, _ = image.shape
27
28      model.setInput(cv2.dnn.blobFromImage(image, size=(300, 300), swapRB=True))
29      output = model.forward()
30
31
32      for detection in output[0, 0, :, :]:
33          confidence = detection[2]
```

```
3.0 0.7795673 car
8.0 0.94995123 truck
3.0 0.5317471 car
8.0 0.9447968 truck
8.0 0.9741377 truck
3.0 0.948382 car
8.0 0.7738217 truck
1.0 0.6333063 person
1.0 0.69034237 person
1.0 0.8215401 person
1.0 0.72127444 person
```

Athula Wickremasinghe I Muon Monitors Data

🔷 **Fermilab**

Simple example of predicting beamXY

Let's see a demonstration of this example in the TSD server

🐝 Fermilab