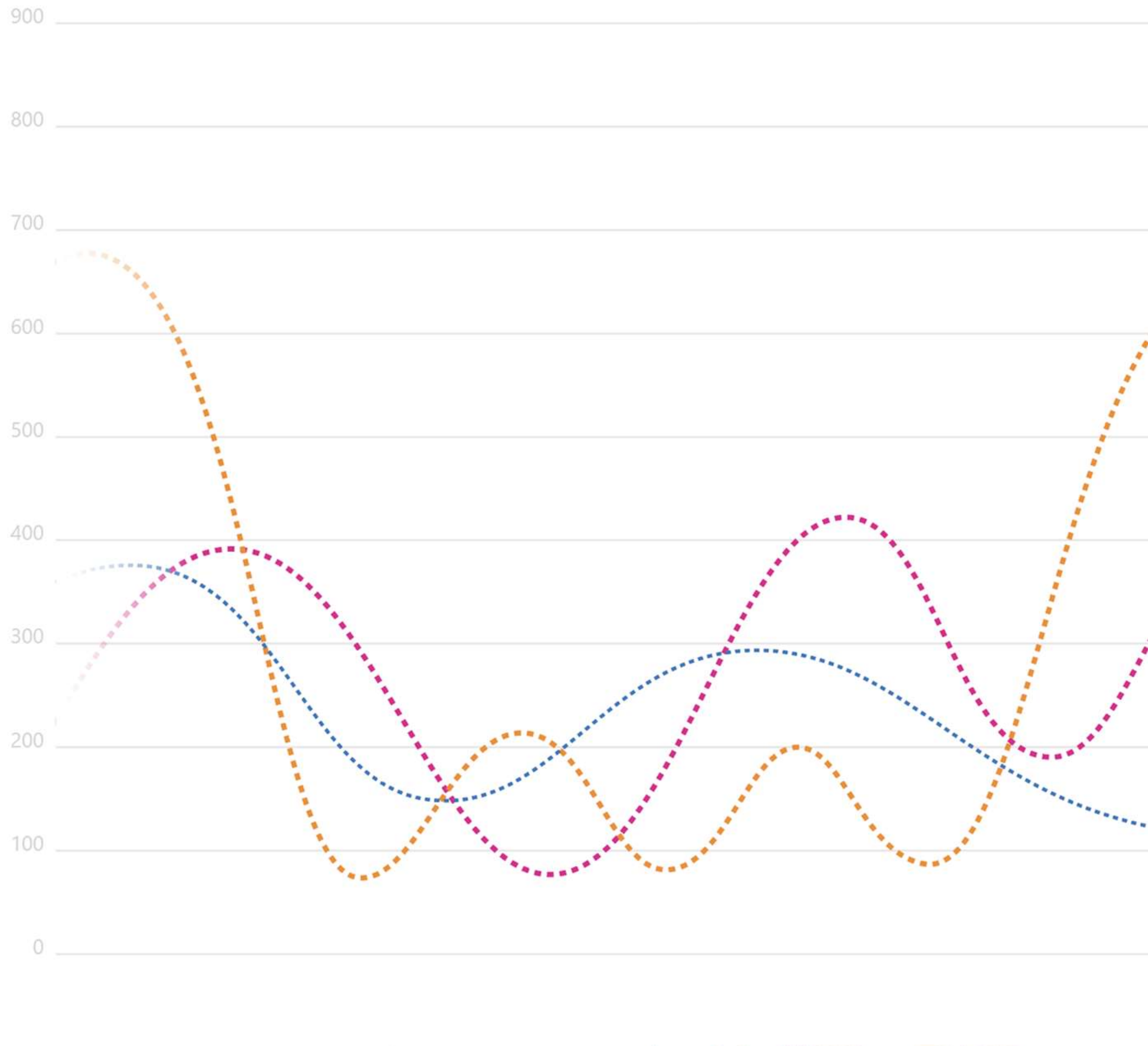
NOAA Fish Detector

Anusua Trivedi, Microsoft
Shreyaan Pathak, Student Intern, Microsoft
Sreya Muppalla, Student Intern, Microsoft
Rahul Dodhia, Microsoft
Jane Wang, Microsoft
Dan Morris, Microsoft
Bridget Ferriss, NOAA
Beth Sanderson, NOAA
Karl Veggerby, NOAA

AI For Good Research Lab (AI4E)



Our Mission

Infuse Data Science and AI to address the world's great challenges

AI for Good
- AI for Humanitarian Action
- AI for Earth
- AI for Accessibility
- AI for Health
- AI for Cultural Heritage

Local Economic Opportunities

Trust in Technology

CELA Initiatives

Introduction

- Maintaining healthy fish populations is vital to US economy—important for commercial and recreational use, integral to our coastal communities, and providing healthy sources of protein.

- NOAA Fisheries scientists are working to find out which species of fish are found in a location for managing sustainable marine and migratory fish populations.

- **Business problem**: NOAA Fisheries would like an automated way to detect fishes and classify the species.

- **Data Science problem**: We propose to create an automated fish detector for NOAA. This would help NOAA detect the fish in a video and label the species of fish.

A little introduction to Deep Learning

Traditional ML Vs DL

Traditional ML requires manual feature extraction/engineering

Deep learning can automatically learn features in data

Feature extraction for unstructured data is very difficult

Deep learning is largely a "black box" technique, updating learned weights at each layer

Why is DL popular?

- DL models has been here for a long time
  - Fukushima (1980) – Neo-Cognitron
  - LeCun (1989) – Convolutional Neural Network

- DL popularity grew recently
  - With growth of Big Data
  - With the advent of powerful GPUs

Deep learning begins with a little function

It all starts with a humble linear function called a perceptron.

$$
\begin{array}{rl}
& \text{weight1} \times \text{input1} \\
& \text{weight2} \times \text{input2} \\
+ & \text{weight3} \times \text{input3} \\
\hline
& \text{sum}
\end{array}
$$

Perceptron:
If sum > threshold: output 1
Else: output 0

Example: The inputs can be your data. Question: Should I buy this car?

$$
\begin{array}{rl}
& 0.2 \times \text{gas mileage} \\
& 0.3 \times \text{horsepower} \\
+ & 0.5 \times \text{num cup holders} \\
\hline
& \text{sum}
\end{array}
$$

Perceptron:
If sum > threshold: buy
Else: walk

These little functions are chained together

- Deep learning comes from chaining a bunch of these little functions together. Chained together, they are called neurons.

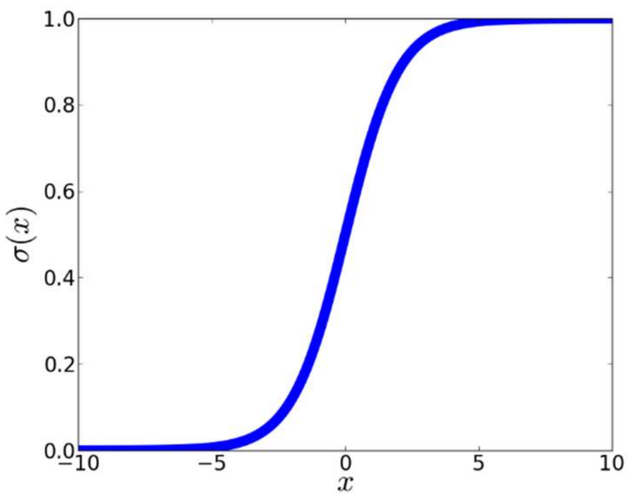
- To create a neuron, we add a nonlinearity to the perceptron to get extra representational power when we chain them together.

- Our nonlinear perceptron is sometimes called a sigmoid.

$$\sigma(\sum_i w_i x_i + b)$$

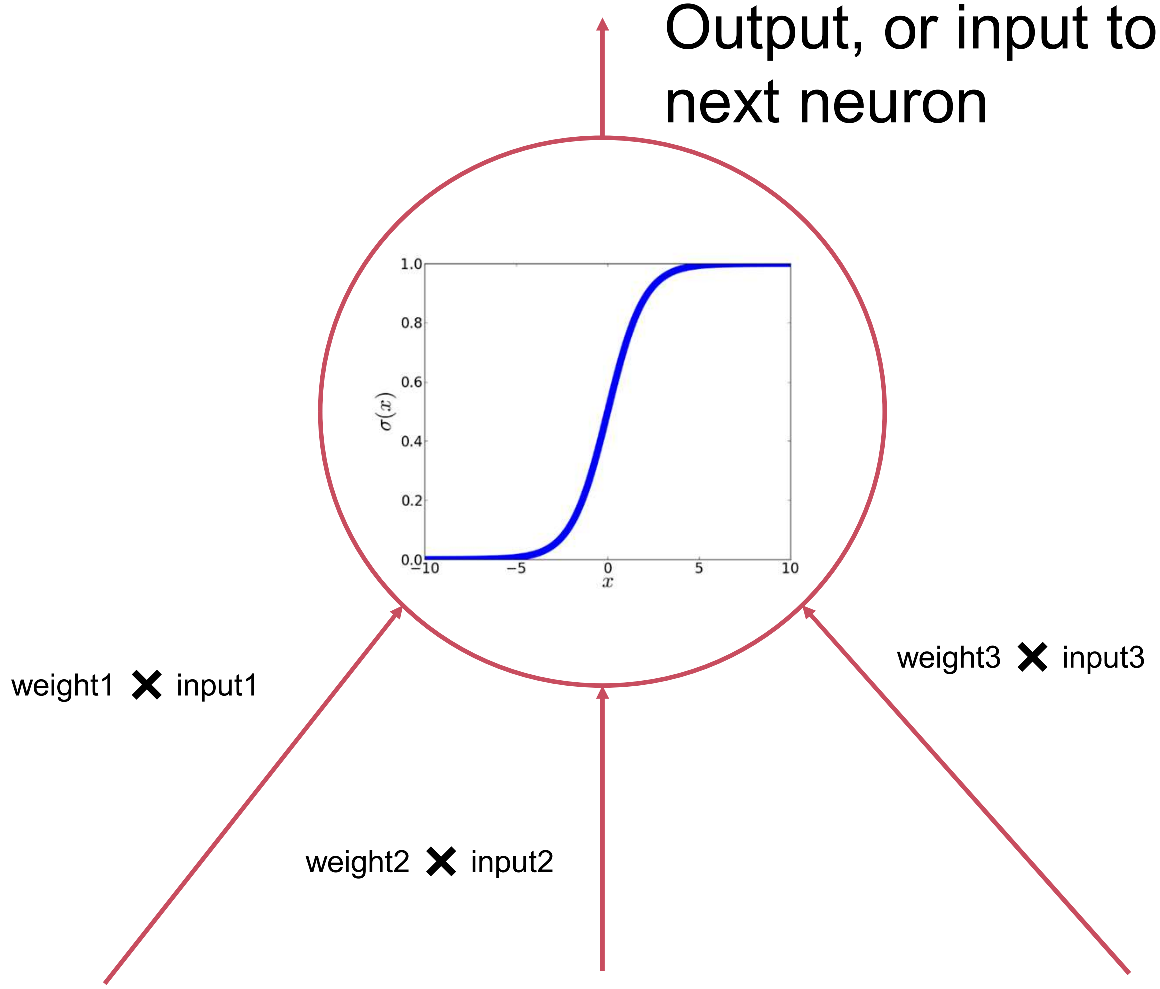
where $$\sigma(x) = \frac{1}{1 + \frac{1}{e^x}}$$

The value b just offsets the sigmoid so the center is at 0.

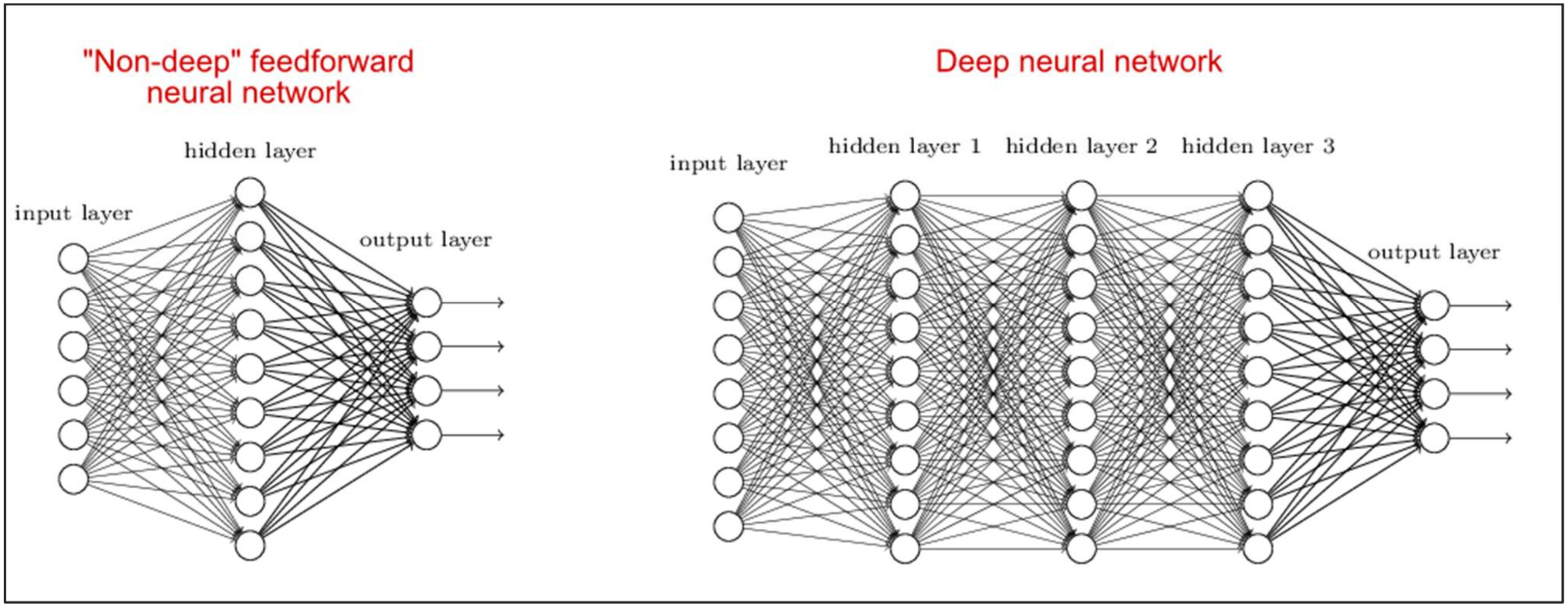


Plot of a sigmoid

Single artificial neuron



Deep Neural Network (DNN)



*http://neuralnetworksanddeeplearning.com/chap5.html

Common DNNs

- ❑ Deep Convolutional Neural Network (DCNN)
  - To extract representation from images
- ❑ Recurrent Neural Network (RNN)
  - To extract representation from sequential data

Microsoft

Computer Vision

Computer Vision Tasks

Image Classification

- Is there a deer in the image?

Object detection

- Where in the image is the deer?

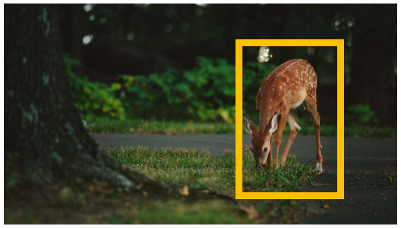


Image segmentation

- Where exactly is the deer? What pixels?

Image Similarity

- Which images are similar to the query image?

Microsoft

Computer Vision using Deep Learning

Deep Neural Network for Image Classification

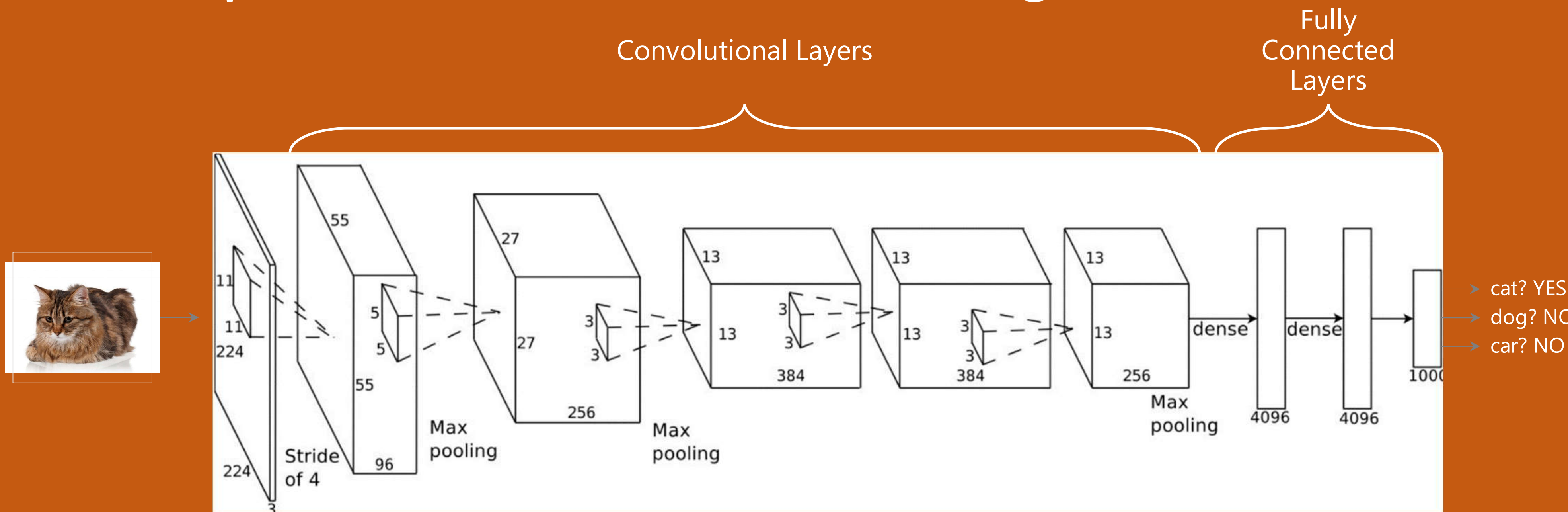
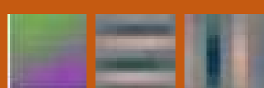
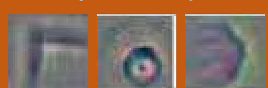
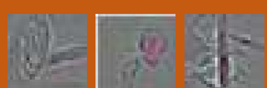


Image → Low-Level Features (lines, edges, color fields, etc.) → High-Level Features (corners, contours, simple shapes) → Object Parts (wheels, faces, windows, etc.) → Complex Objects & Scenes (people, animals, cars, beach scene, etc.)

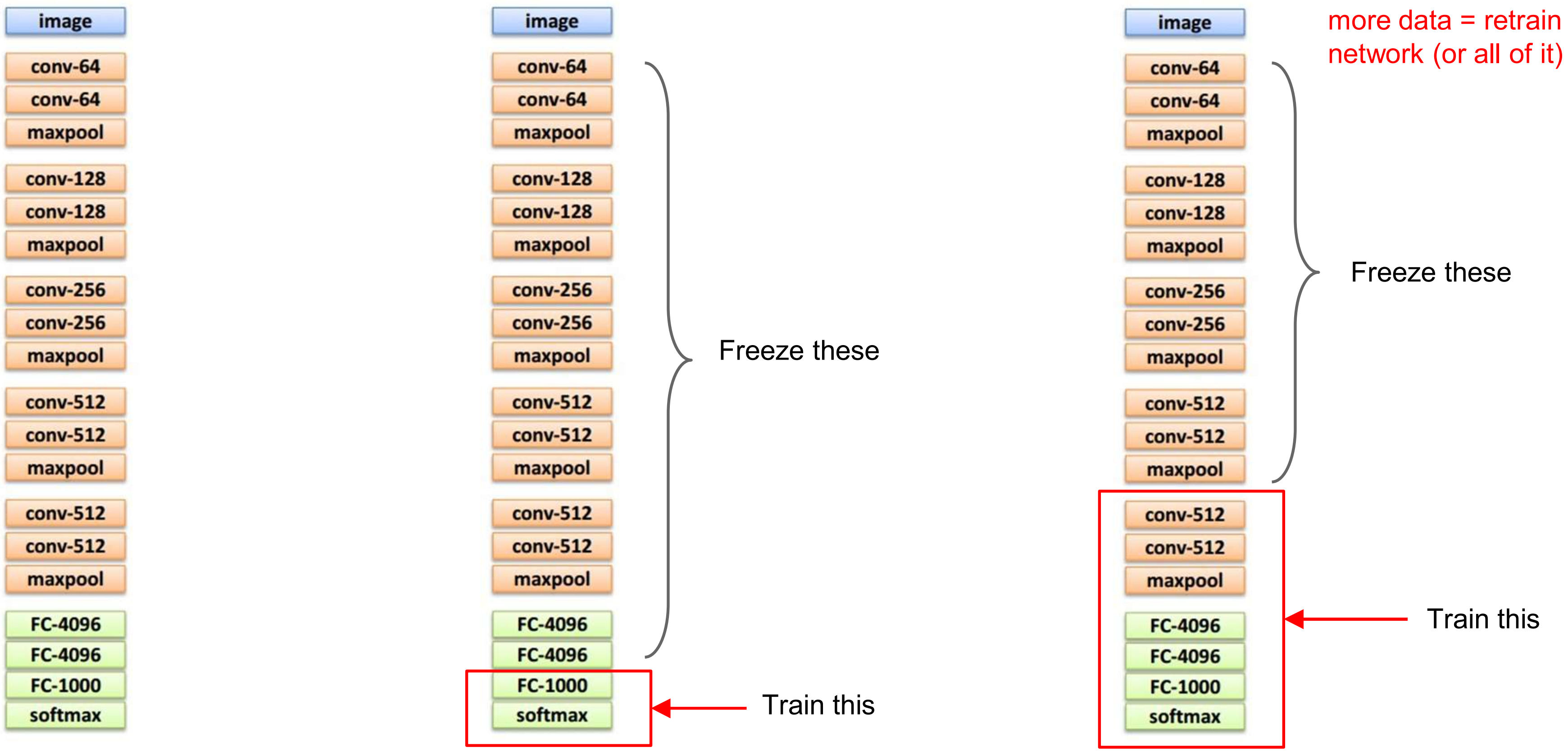
Transfer Learning

1. Train on Imagenet

2. Small dataset: **feature extractor**

3. Medium dataset: **finetuning**

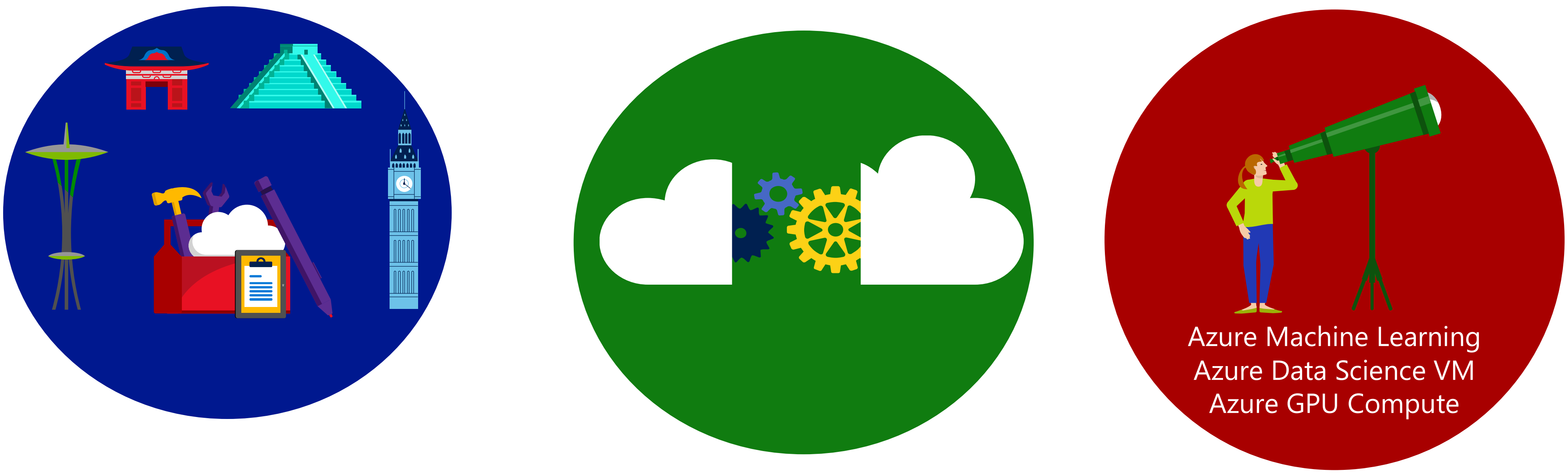
more data = retrain more of the network (or all of it)



*Andrej Karpathy's recent presentation

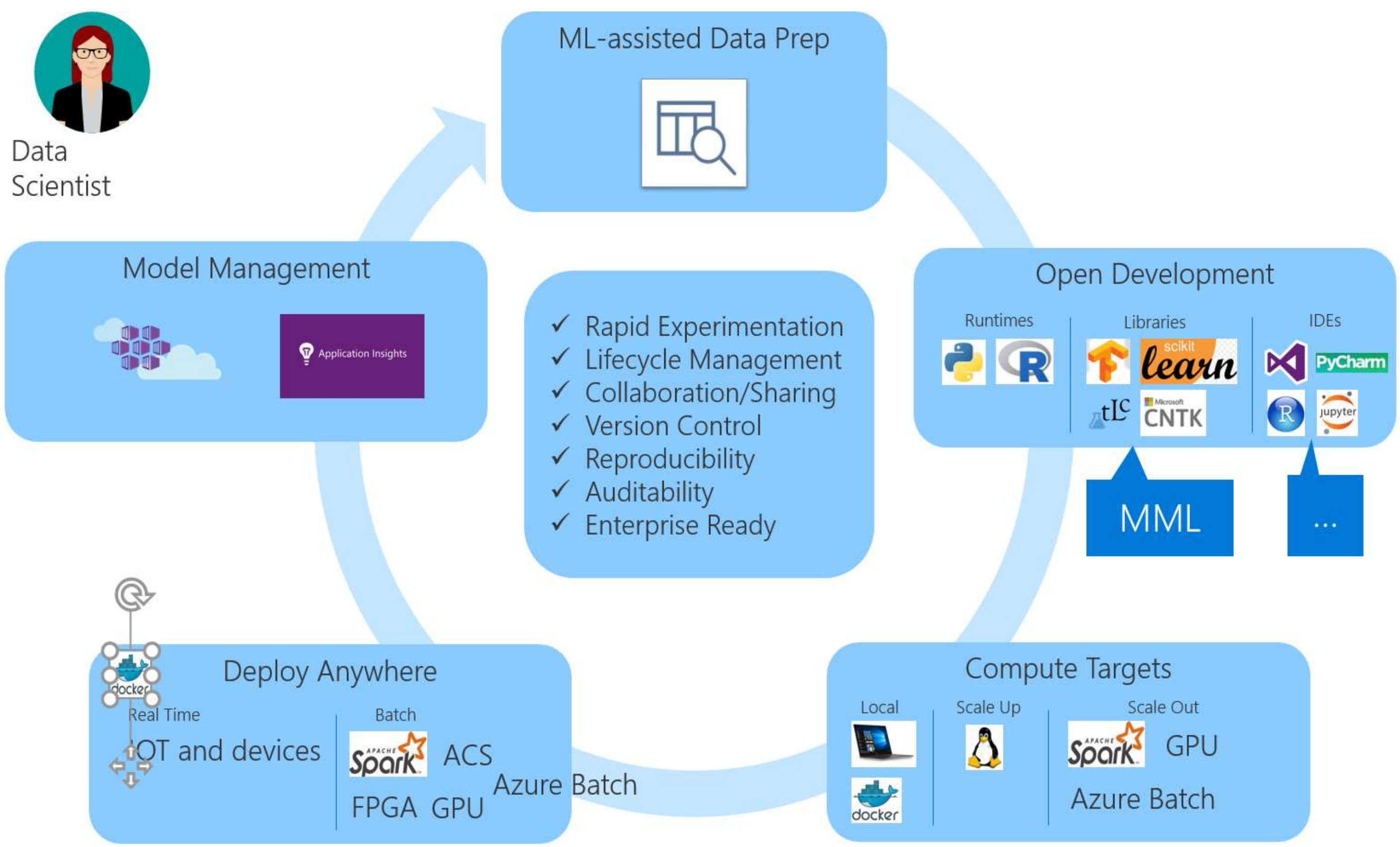
Microsoft AI Platform

Azure End-2-End Customer Solution



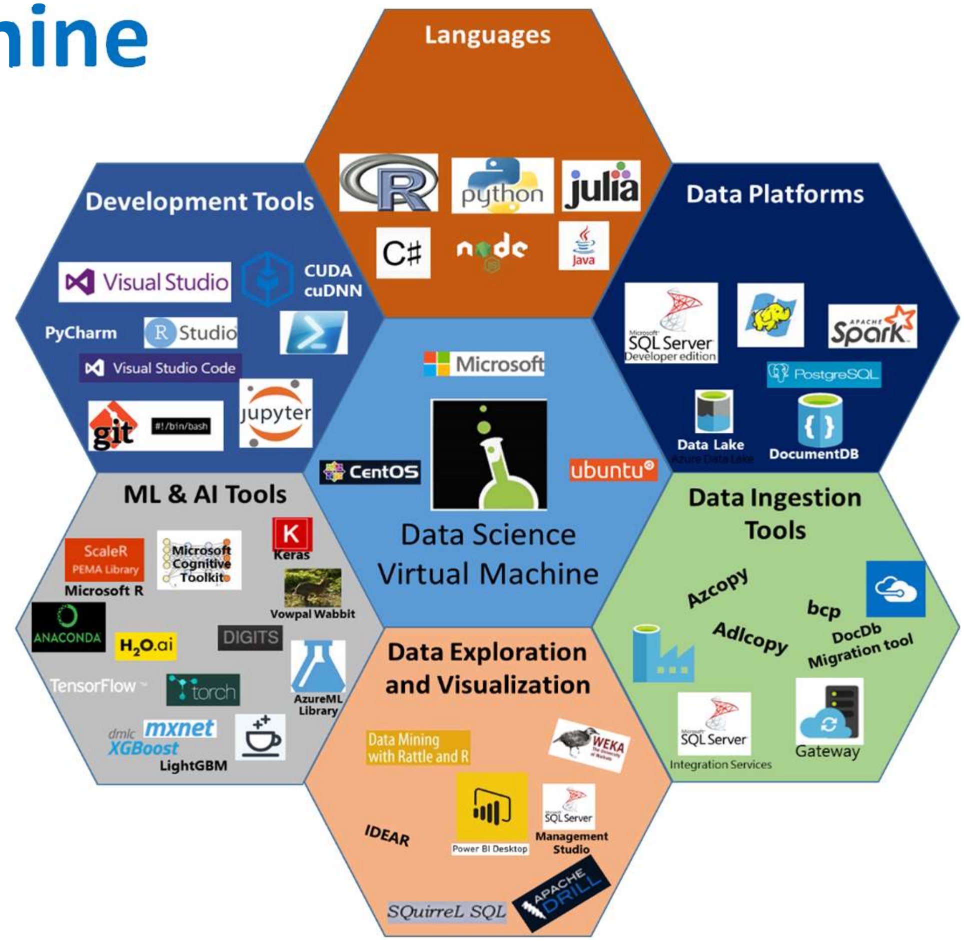
Your Data + Microsoft AI Platform Components = Intelligence In Your Apps and Data Services

Microsoft AI Platform



Data Science Virtual Machine

Running our Image Processing Pipeline in a Data Science Virtual Machine (DSVM) with Deep Learning products pre-stalled



Objective

The ***goal*** of this fish detection project:

- ❑ The NOAA scientists have been collecting underwater videos from various locations around Puget sound area. NOAA wants to identify a fish in underwater videos first, and then classify the fish for species population management.

- ❑ However, they are doing the whole process manually (i.e. a person goes through each of the videos manually trying to detect a moving fish).

- ❑ Automating this curation process would reduce thousands of hours of work the small team spends each day.

The ***objective*** is to build a precise object detection model to detect the fishes in the underwater videos.

Methods

- ❑ Prior to Microsoft, NOAA Fisheries collaborated with a UW professor on this project. However, they were not able to make much progress on this problem

- ❑ We received about ~2000 annotated fish images from the UW collaborator
  - ➢ The images were from videos with one background location only

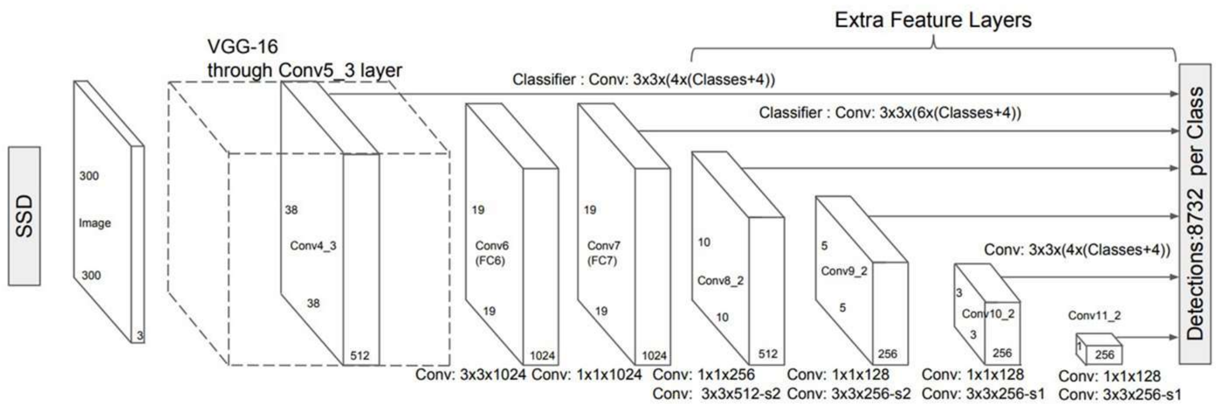
- ❑ We decided to use a neural network-based approach for detecting the fishes and trained an image-based object detector for detecting the fish.

- ❑ An image object detector localizes the fish by predicting the bounding boxes and detection confidences.

- ❑ Among the state-of-the-art detectors, we choose the MobileNet single-shot detectors (SSD) for their high efficiency, high latency and portability.

Methods

- ❑ In the SSD-MobileNet model:
  - ❑ A backbone convolutional neural network is used to extract the feature maps of the input image at different scales.
  - ❑ In each feature map, several default boxes of different aspect ratios at each location are evaluated for their offset from the actual bounding boxes of fish and their detection confidences for all fishes.



Reference:
https://arxiv.org/pdf/1512.02325.pdf

Pros/Cons

- ❑ The only caveat is overfitting – we focused a lot on the hyperparameter tuning to avoid any such training overfitting

- ❑ For real-time inferencing, we created a multi-threaded schedular which calls the trained fish-detector-model on every frame of a video

Challenges with the V1 Prototype Model

The model does not generalize well due to the following reasons:

- ❑ videos are very blurry
- ❑ Varying background (moving grass, metal grill, murky ground etc.)
- ❑ Different camera filter colors
- ❑ Some fishes are translucent and are very similar to background (camouflaged)

Thus, the detection results are usually very noisy, which leads to high false positives

V2 Phase

- ❑ We generated ~68,050 images from 200K videos with different backgrounds (balanced dataset on background).

- ❑ AI4E Vendor (iMerit) helped us annotate these 68K images with bounding boxes (fish as the only category).

- ❑ V2 Phase goal is to reduce the high false positives on various backgrounds

- ❑ V2 Phase has 2 models
  - model V1 which tries to identify any non flat fishes
  - model V2 tries to identify only flat fishes near the ground (hard to locate for humans)

*currently V1 works well

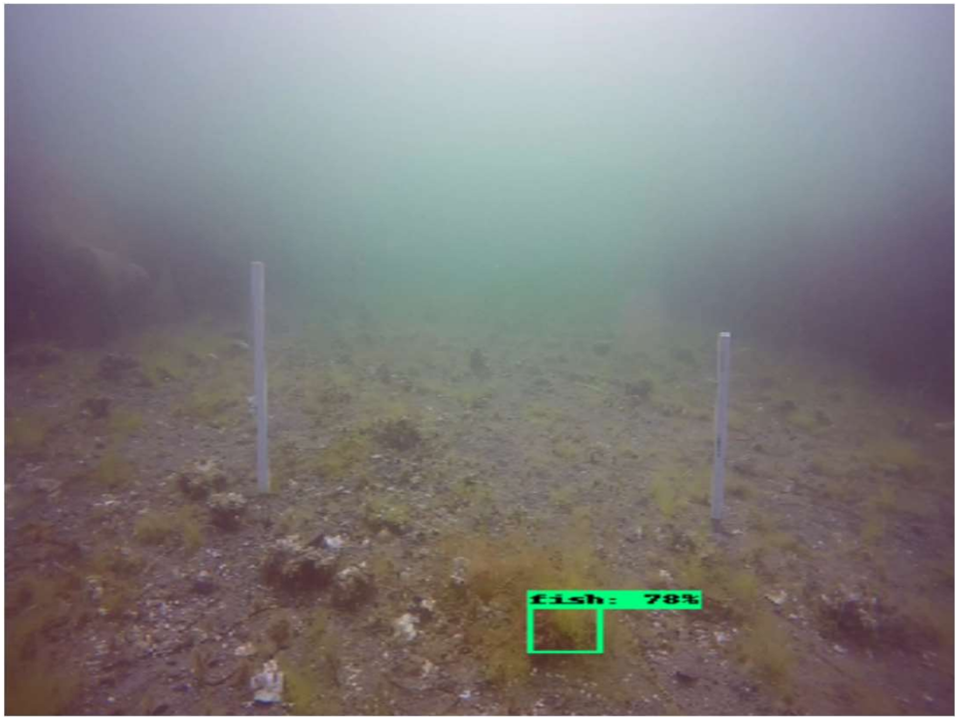
– "AUC": "0.80847", "Precision": "0.85731", "Recall": "0.70907"

*V2 has a high false positive rate, needs improvement

– "AUC": "0.78869", "Precision": "0.68925", "Recall": "0.79105"

DEMO


shiner_perch: 79%
shiner_pe


fish: 78%

Setup/Prerequisites

- Python: 3.5.5
- CUDA: 10.0
- CUDNN: 7
- Creating and activating a new python environment
- req.txt
- export PYTHONPATH=$PYTHONPATH:<user_path>/models/research:<user_path>/models/research/slim

https://github.com/antriv/NOAA_Fish_Detection/blob/master/README.md

Fish Detection

Input

- Multiple videos in a directory
- Single Video

Output

- One CSV file for each video file in the CSV directory
- One IMAGE folder for each video in the IMAGE directory

Fish Detection

Multiple Video

- make directories
- run following command in terminal
- python noaa_imerit_2_main_inference_multiple_videoUse.py

--pathVideo <path to video directory(all video files) >

--pathCSV

--pathIMG

Single Video

- make directories
- run following command in terminal

python noaa_imerit_1_main_inference_single_videoUse.py

--pathVideo

--pathCSV

--pathIMG

GUI

- Install X-Server locally and activate when connecting to VM
- Run following command in terminal
- python <dir_path>/models/research/object_detection/noaa_imerit_OD_inference.py

Files Needed

- noaa_imerit_1_main_inference_single_videoUse.py

- Used to detect fish for one video

- noaa_imerit_2_main_inference_multiple_videoUse.py

- Used to detect fish for multiple videos

Files Needed (backend dependency code)

- noaa_imerit_OD_inference_mainUse.py

- Used to generate a csv of frame numbers(ie timestamps) for when fish are detected for a single video

- noaa_imerit_main_inference_single_video.py

- Used to detect fish for one video

- noaa_imerit_framesUse.py

- Used to generate a folder with images generated by both v1 and v2 models for a single video

Files Needed

- noaa_imerit_main_condition_detection.py
- Used to produce two CSVs:

- a copy of the original csv with bounding box ids labeled
- a csv with timestamps of bounding box ids

How to run:

```
python noaa_imerit_main_condition_detection.py
--pathCSV
<dir_path>/Project_NOAA_imerit/outimg/csvfiles/<csv_name>
--threshold <threshold>
```

CODE WALKTHROUGH

GITHUB:
https://github.com/antriv/NOAA_Fish_Detection

Final Outcome

- We have 3 types of inferencing scripts
  - single video gui based detection for demo/human-inspection purposes
  - terminal based detection for single video
  - terminal based detection for multiple videos
- we produced images out of videos
  - images contains only the frames in which fishes were detects
- this significantly reduces the stress on labeling fish videos
- these models still have some false positive predictions

Thank you.

AI for Good Research Lab