Deep Learning

The Fundamentals

Benjamin Akera

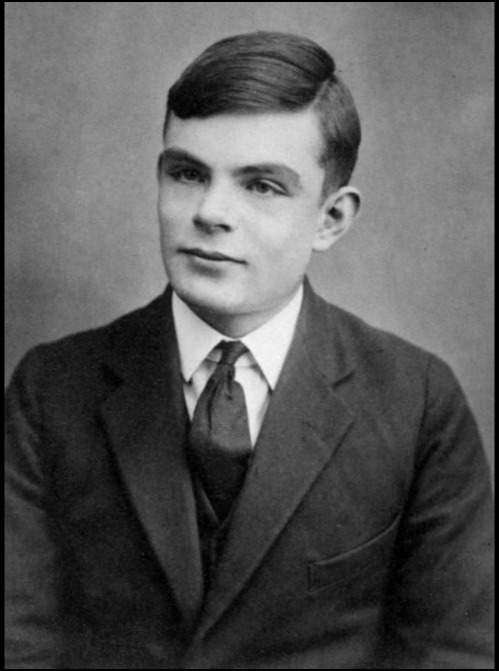
MILA - Quebec AI Institute

Artificial Intelligence
&
Machine Learning

Let's start from the very beginning

I propose to consider the question "can machines think?"

-Alan Turing (1950)



What is Artificial Intelligence?

- Artificial Intelligence is the Intelligence of machines and the branch of computer science that aims to create it

Machine Learning

- Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed

How do the machines learn?

Supervised Learning

Learning with a labelled training set

- Email Spam detector with training set of already labelled emails

Unsupervised Learning

Discovering patterns in unlabelled data

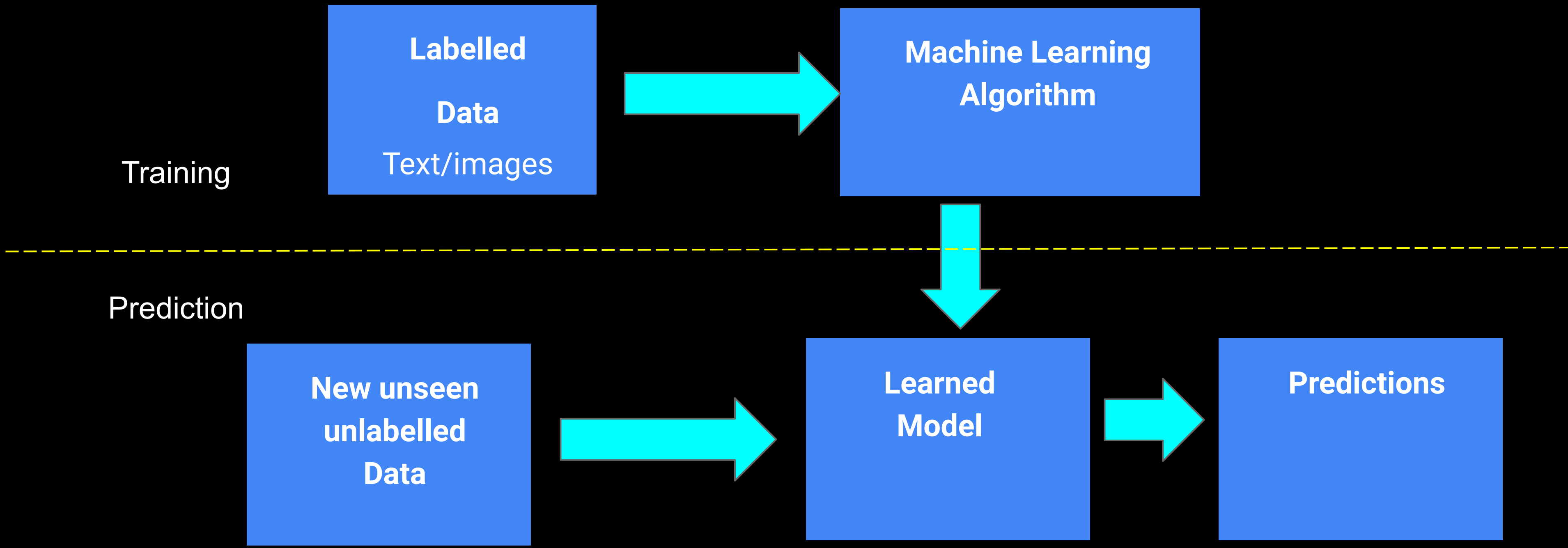
- Cluster similar documents based on the text content
- Word Embeddings for NLP

ReInforcement Learning

Learning based on feedback or reward

- Learn to play chess by winning or losing
- Alpha GO, AGZ

Process of Supervised Machine Learning



What Machine Learning Can Do? (input to output mappings)

A simple way to Look at Supervised Machine Learning

Input (X)	Response (Y)	Application
Picture	**Are there Human faces? (0 or 1)**	**Photo Tagging**
Loan Application	**Will they repay the loan? (0 or 1)**	**Loan approvals**
Ad + User Information	**Will user click an ad?**	**Targeted online advert**
Audio Clip	**Transcript of Audio clip**	**Speech recognition**
English Sentence	**French Sentence**	**Language translation**

What is Deep Learning?

- Deep Learning is a Supervised Machine Learning technique inspired by the structure and function of the brain, mimicking it with components called **Artificial Neurons**

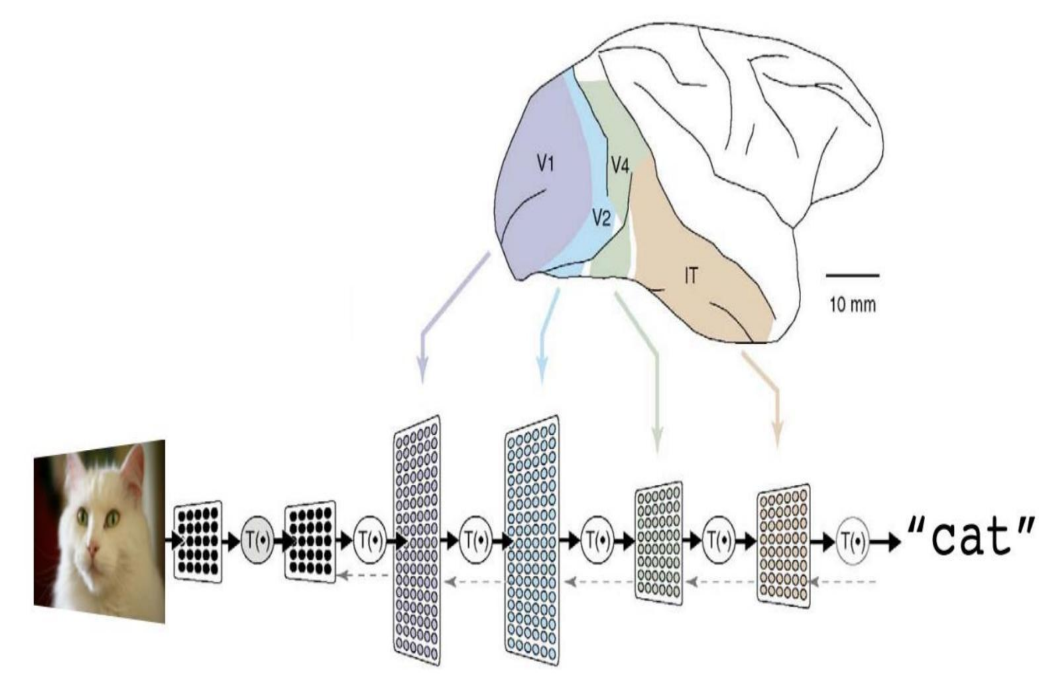
Our focus will be on Deep Learning

Well said Leo, well said

Deep Learning

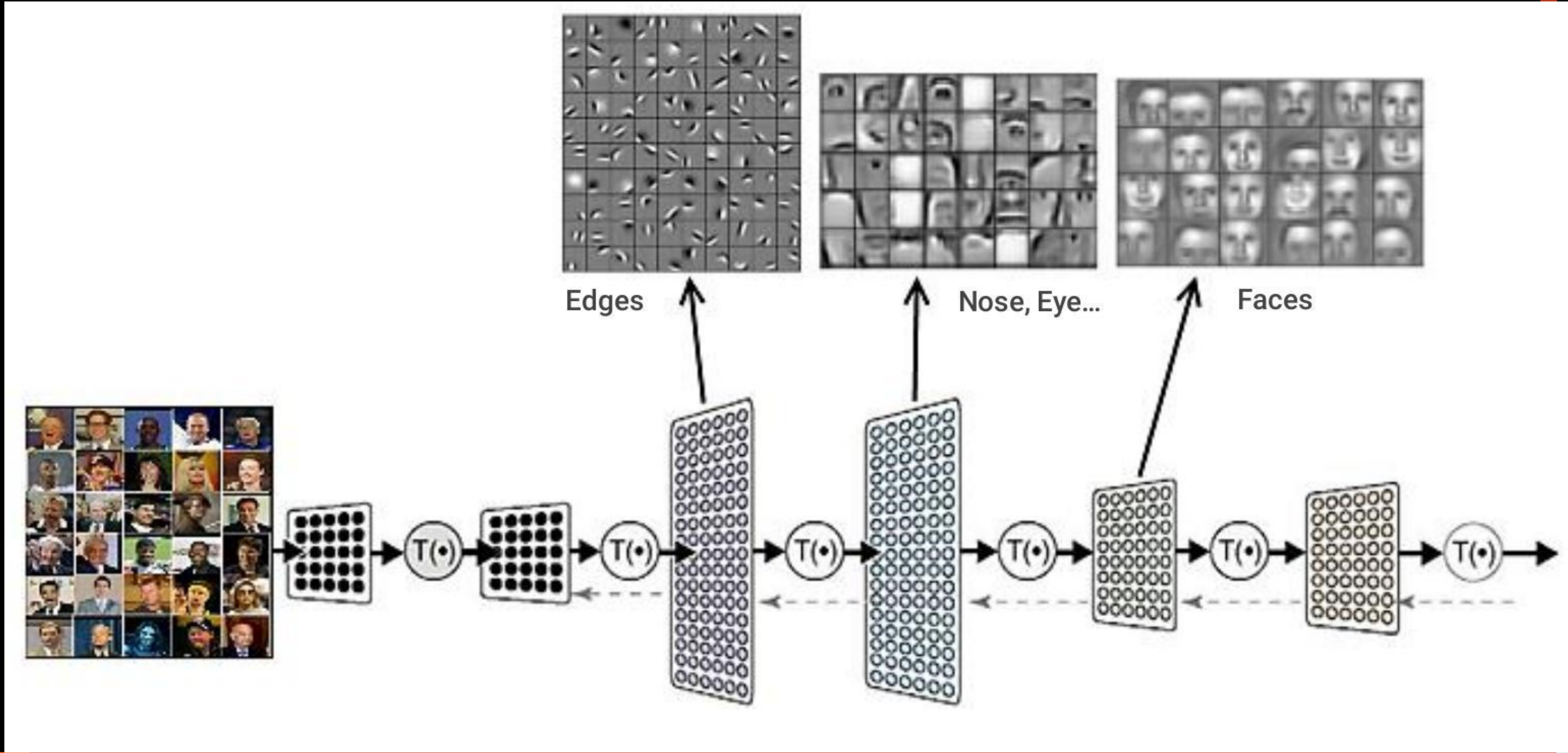
A fancy term for a Neural Network with many hidden layers.

What is Deep Learning?



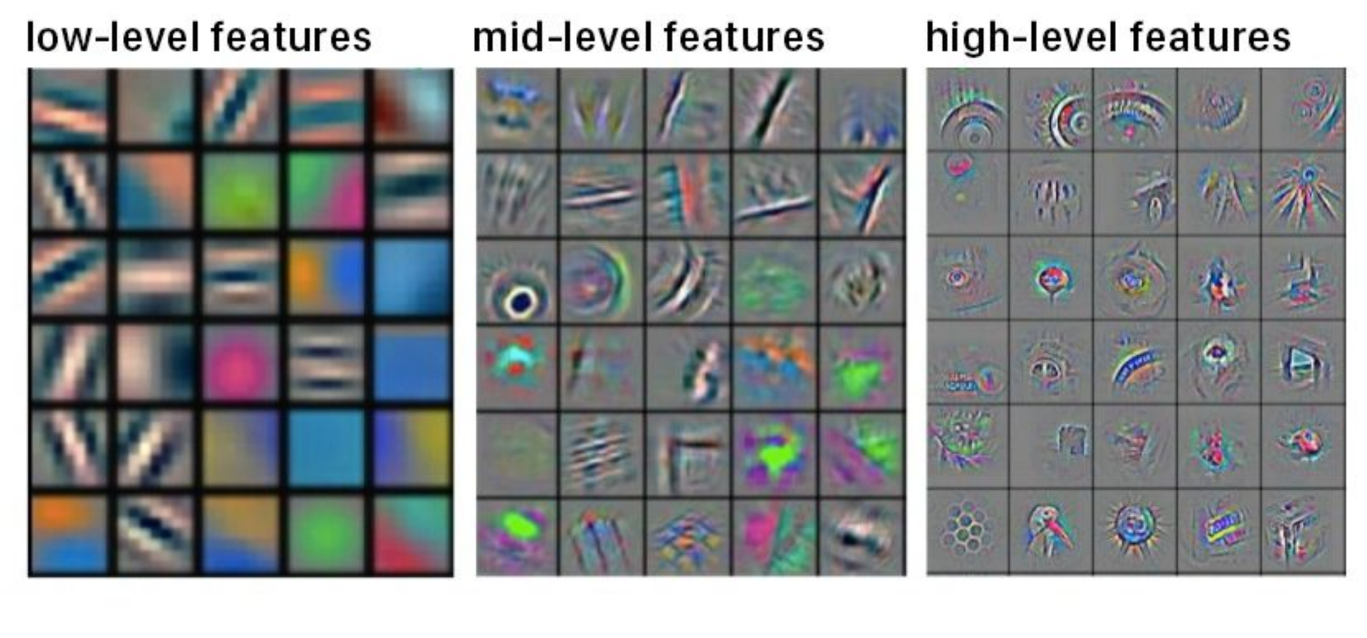
- A deep Neural Network consists of a hierarchy of layers.
- Each Layer transforms the input data into more abstract representations eg. (edge -> nose -> face)
- The output layer combines all these features to make predictions

What did it learn? ...let’s take a closer look



Why Deep Learning

- Hand Engineered features are time consuming and not scalable in practice
- Can We learn the **Underlying features** directly from data?



Why Deep Learning?

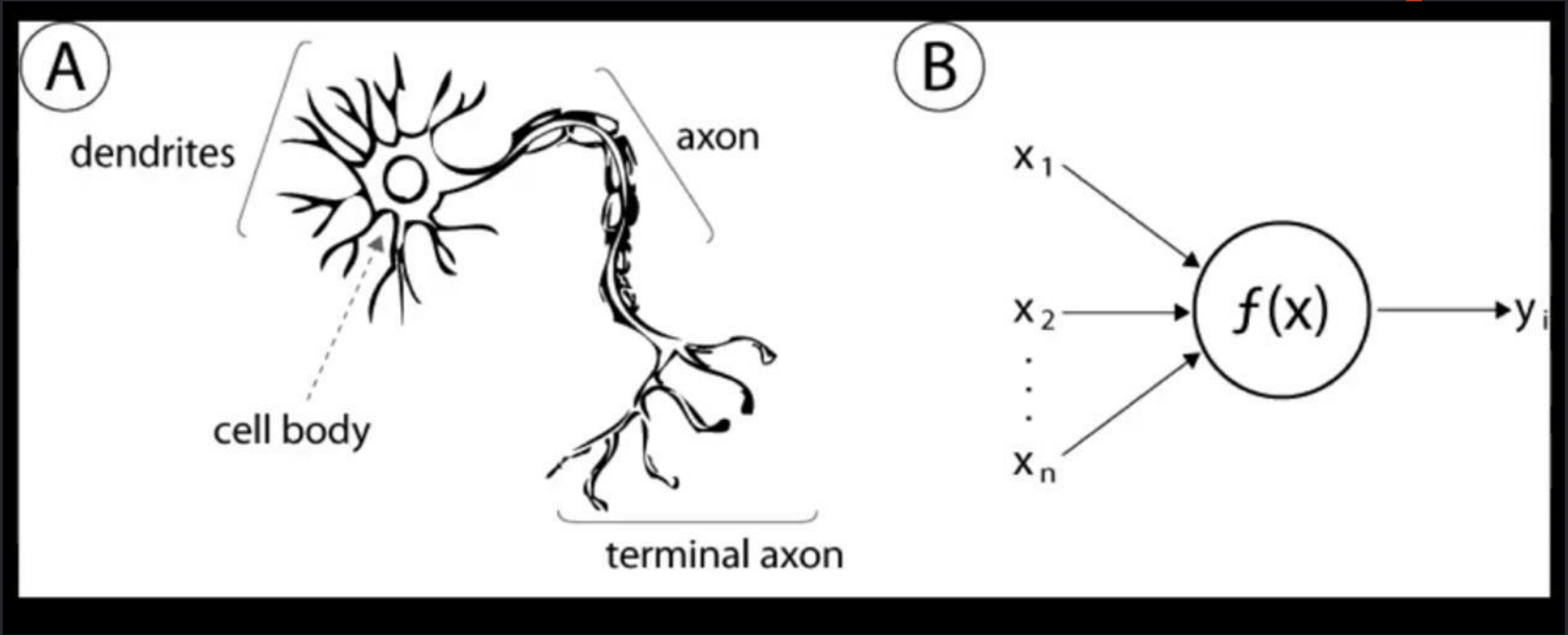
1. Big Data
    - Larger Datasets
    - Easier collection & storage

2. Hardware
    - Graphical Processing Units

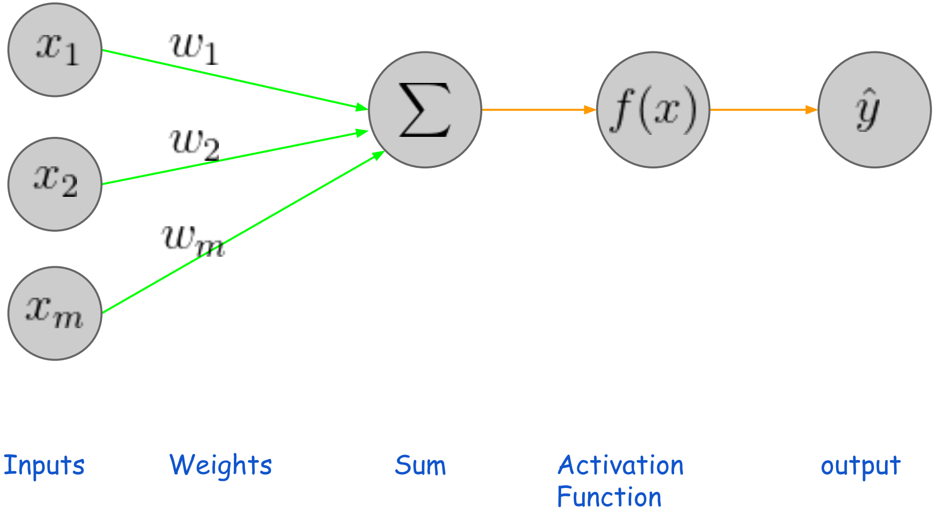
3. Software
    - Improved techniques
    - New Models
    - Tool boxes

Part 1: Mathematical Intuition

Biological Neuron vs Artificial Neuron



The Perceptron: Forward Propagation



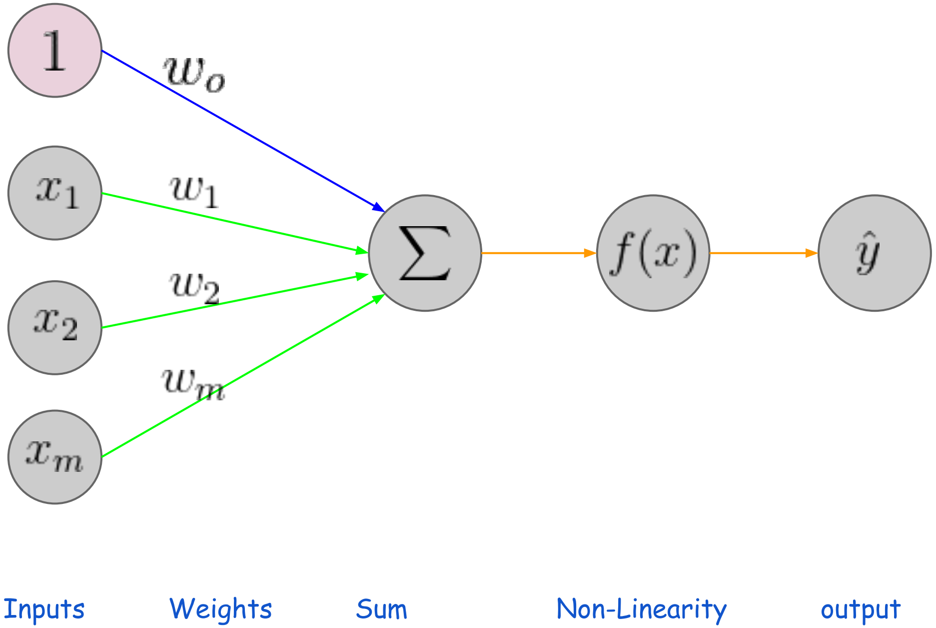
Linear combination of inputs

output

$$\hat{y} = f\left(\sum_{i=1}^{m} x_i w_i\right)$$

Non Linear
Activation function

The Perceptron: Forward Propagation



Linear combination of inputs

output

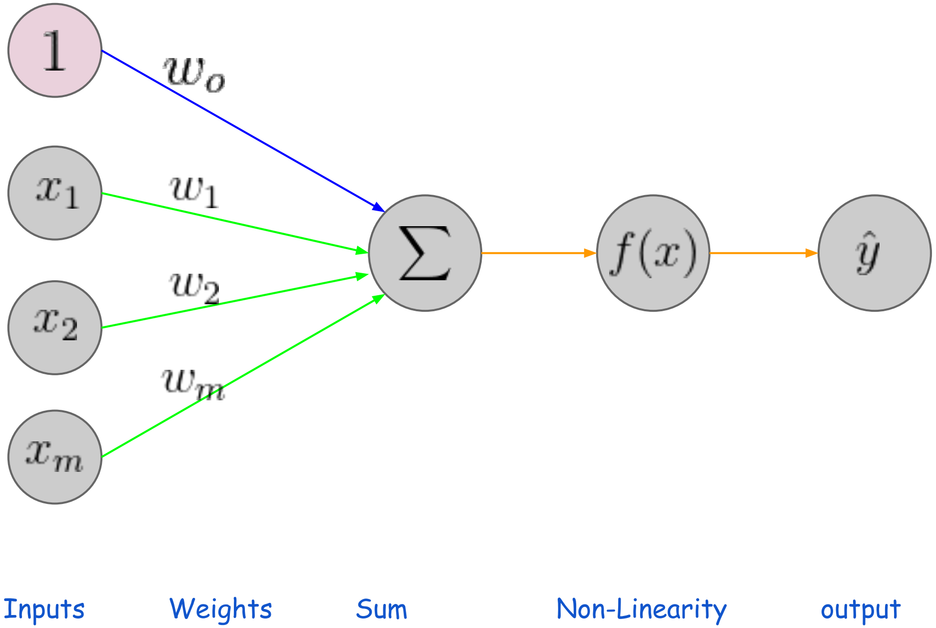
$$\hat{y} = f\left(w_o \sum_{i=1}^{m} x_i w_i\right)$$

Non Linear
Activation function

Bias

The **bias value** allows you to shift the activation function to the left or right, which may be critical for successful learning.

The Perceptron: Forward Propagation

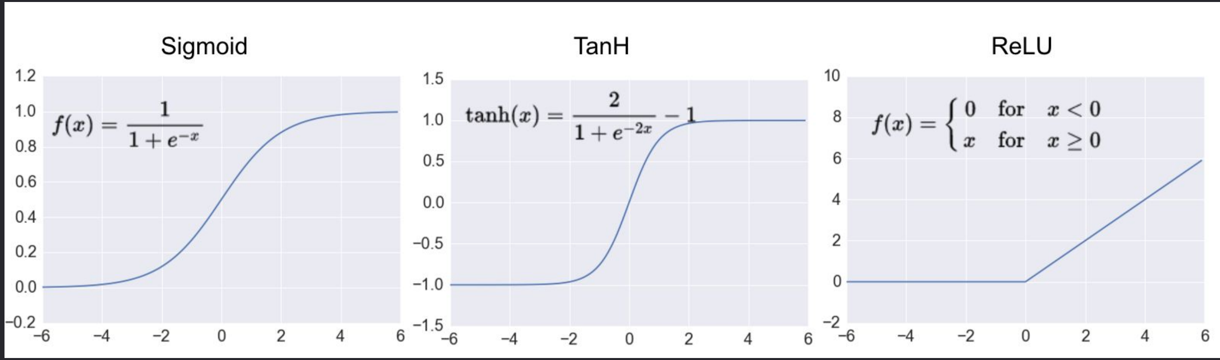


$$\hat{y} = f\left(w_o \sum_{i=1}^{m} x_i w_i\right)$$

$$\hat{y} = f(w_o + X^T W)$$

$$where : X = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_m \end{bmatrix} \; and \quad W = \begin{bmatrix} w_1 \\ . \\ . \\ . \\ w_m \end{bmatrix}$$

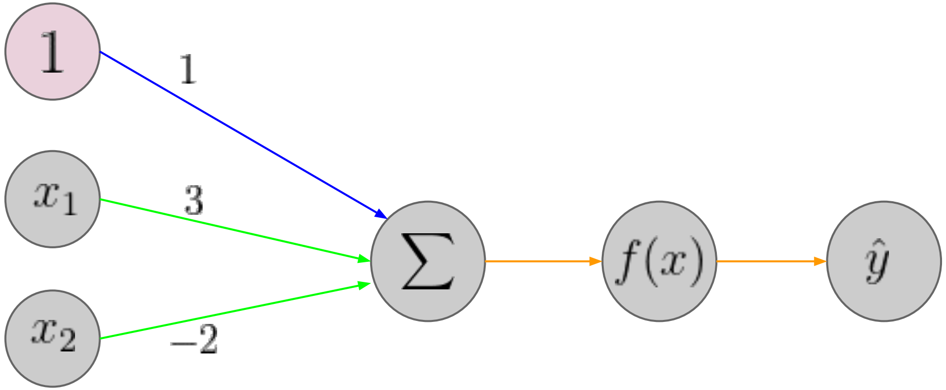
Common Activation Functions

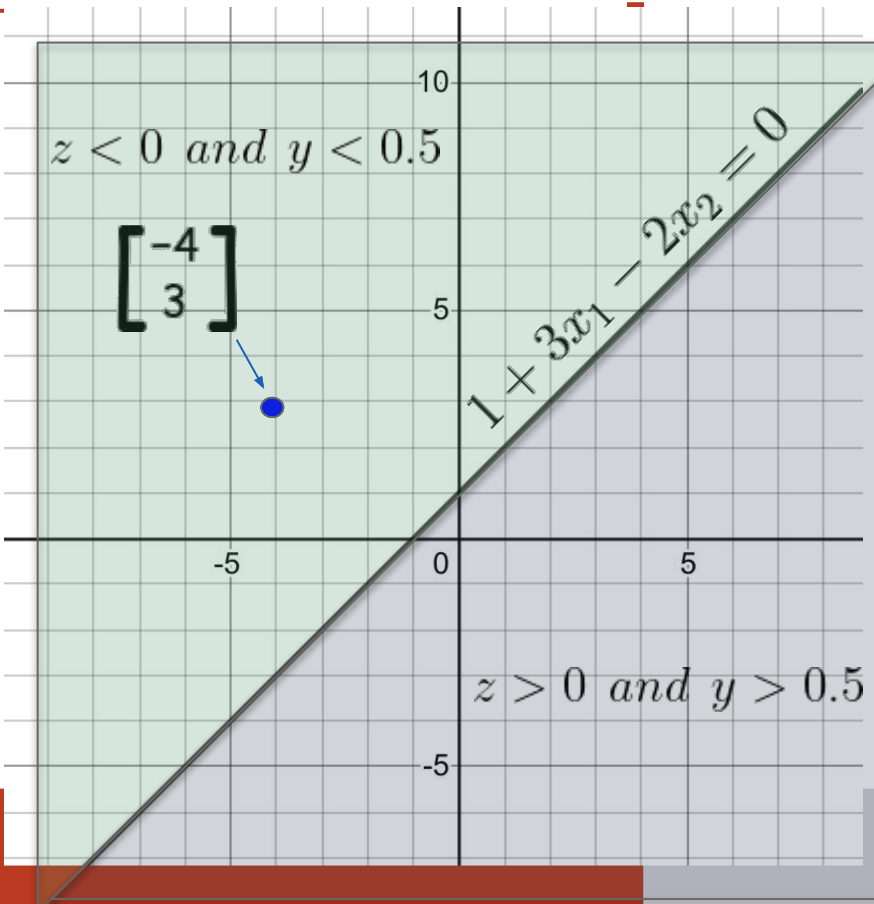


Most Neural Networks use **ReLU - max(0,x)** Nowadays for hidden layers, since it trains much faster.

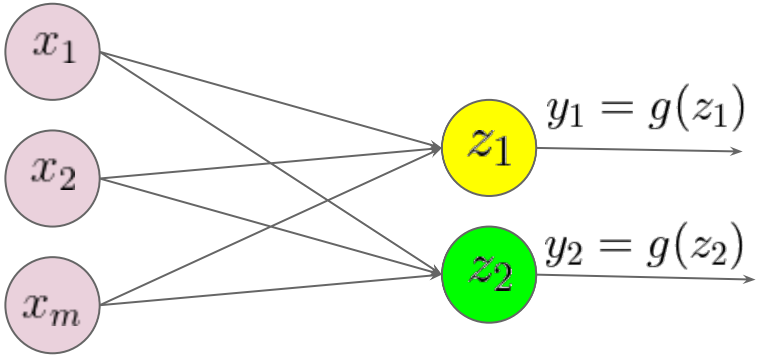
Non linearity is needed to learn more complex (non linear representations of data, otherwise, the NN would just be a linear function

The Perceptron Example1: Single Output



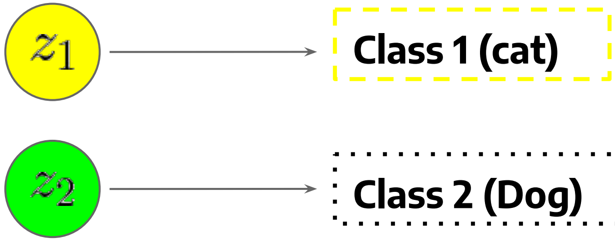


The Perceptron Example2: Multiple Outputs

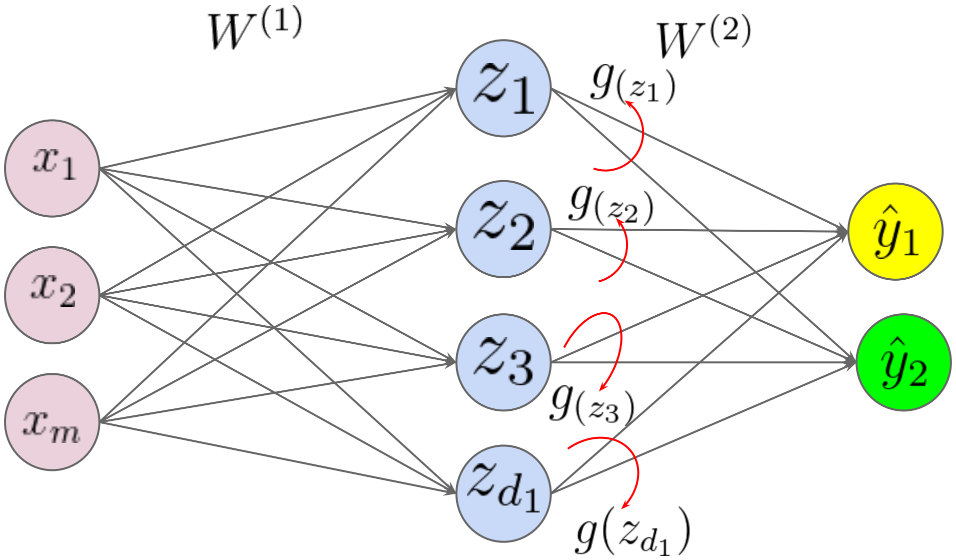


$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j w_{j,i}$$

Since all inputs are "densely" connected to all outputs, these layers are called **Dense** layers



Single Layer Neural Network

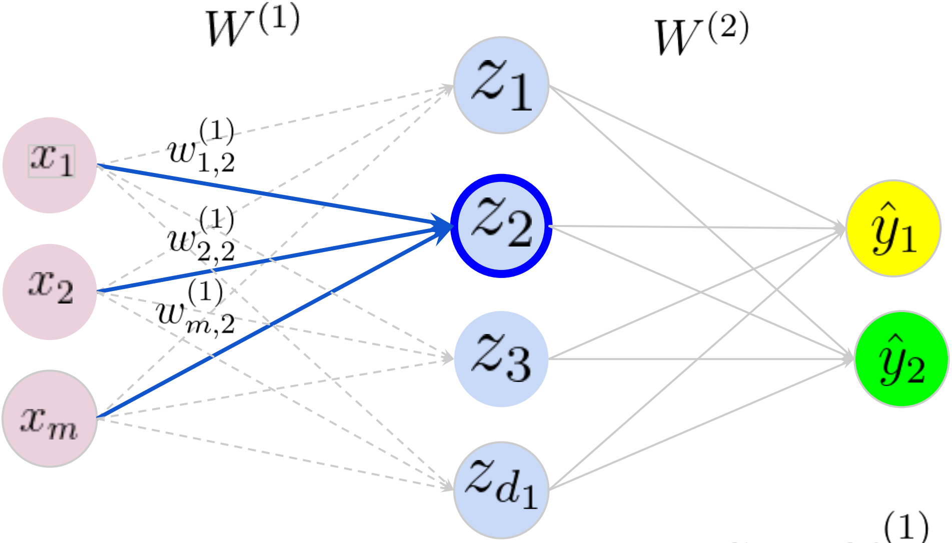


$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j w_{j,i}$$

$$\hat{y}_i = g\left(w_{0,i}^{(2)} + \sum_{j=1}^{d_1} g(z_j) w_{j,i}^{(2)}\right)$$

inputs **hidden** **Final outputs**

Single Layer Computation

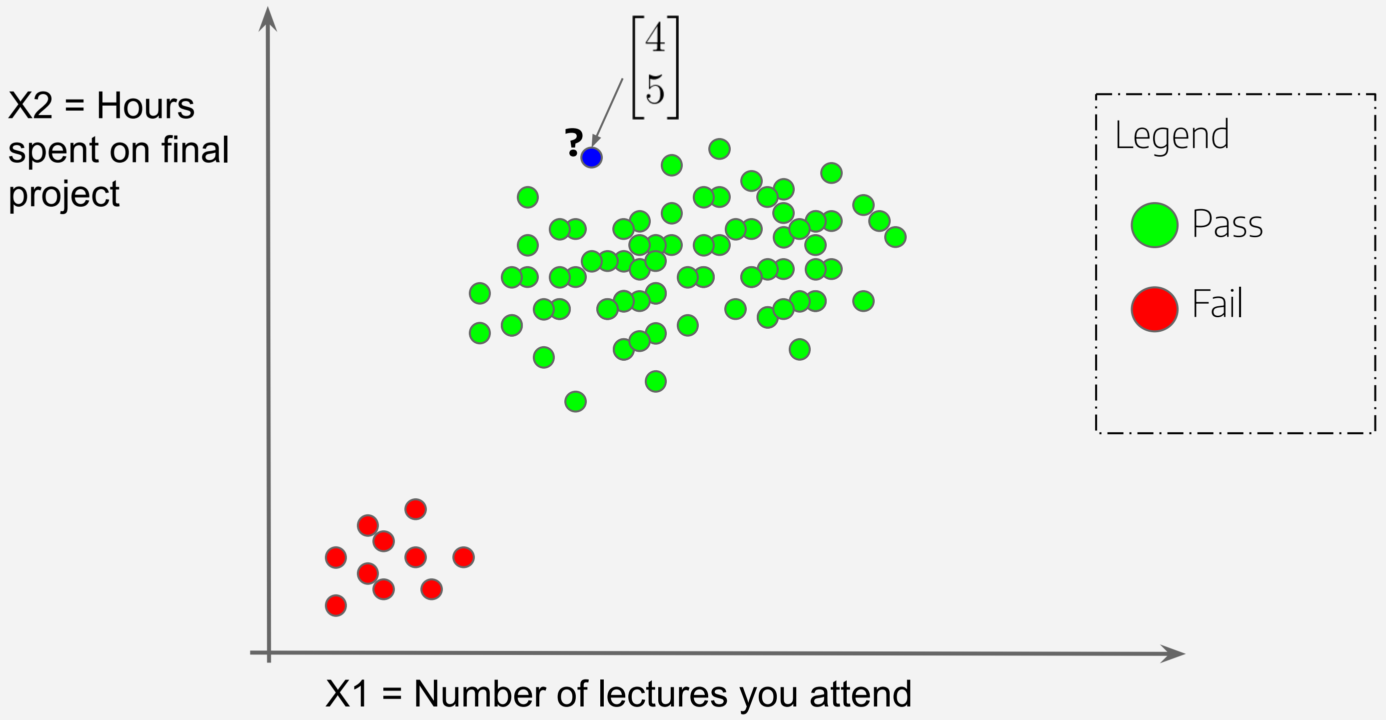


$$z_2 = w_{0,2}^{(1)} + \sum_{j=1}^{m} x_j w_{j,2}^{(1)}$$

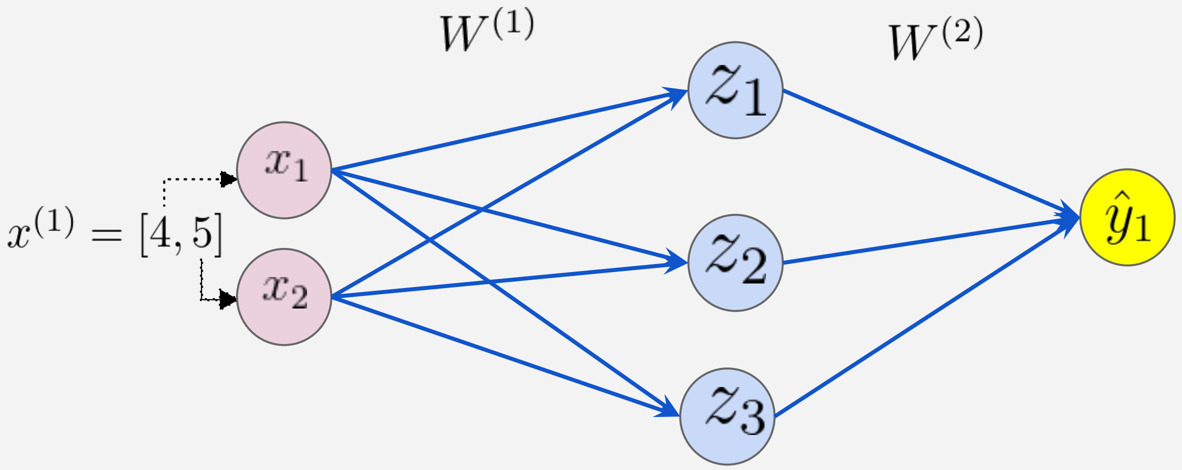
$$= w_{0,2}^{(1)} + x_1 w_{1,2}^{(1)} + x_2 w_{2,2}^{(1)} + x_m w_{m,2}^{(1)}$$

Part 2: Neural Networks In Action

Example Problem: Will a student Pass this class?



Forward Pass through A Neural Network



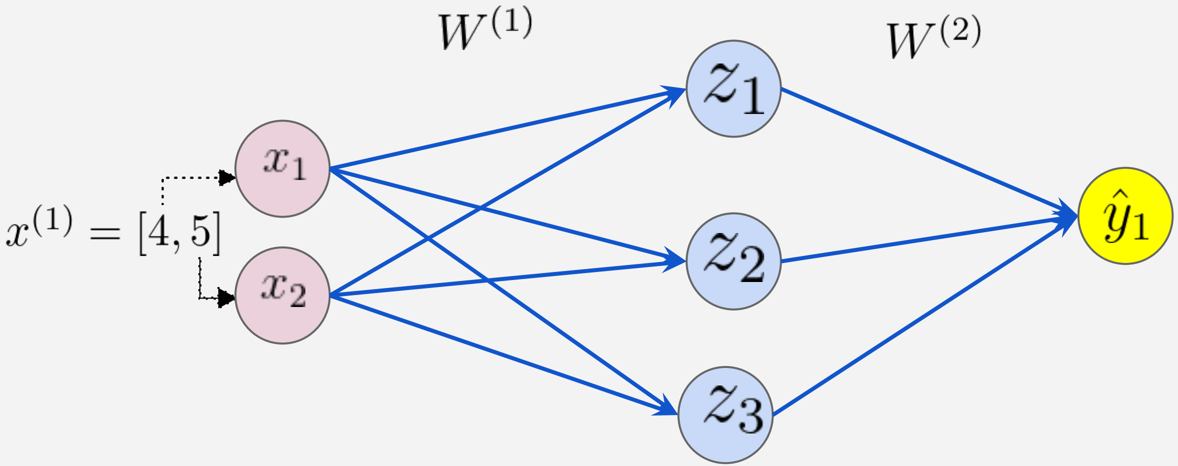
Predicted: 0.1

Actual: 1

Qn: Why did it fail?

Quantifying Loss $(\mathcal{L})$

The **Loss** of the network measures the cost incurred from incorrect predictions



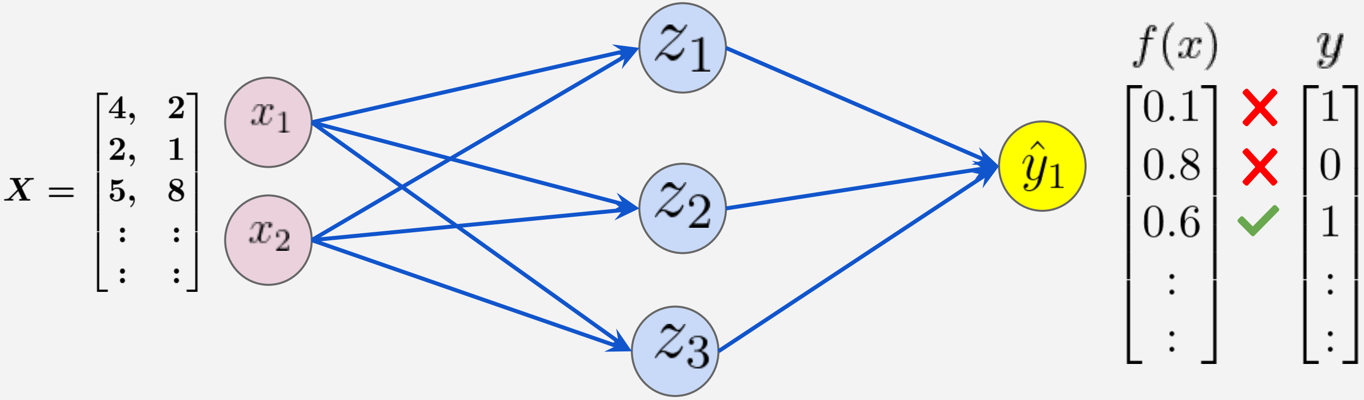
Predicted: **0.1**

Actual: **1**

Loss : $$\mathcal{L}(\underbrace{f(x^{(i)}; \mathbf{W})}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}})$$

Empirical Loss: $J(\mathbf{W})$

The **Empirical Loss** measures the total loss over our **entire dataset**



Empirical Loss

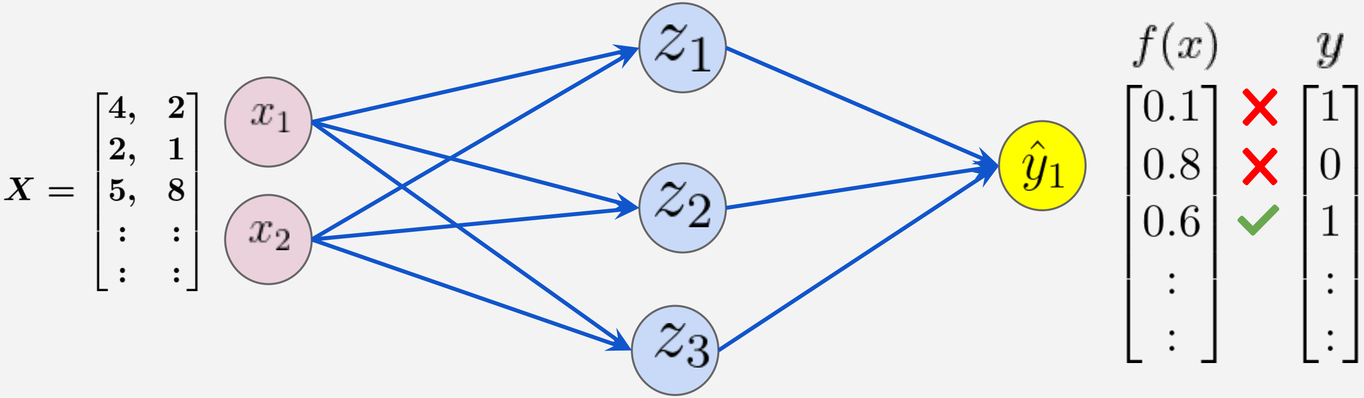
: Also known as:

- Cost function
- Objective Function
- Empirical risk

$$J(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}; \mathbf{W}\right), y^{(i)}\right)$$

Binary Cross Entropy Loss (BCE Loss)

BCE Loss can be used with models that output a probability between 0 and 1

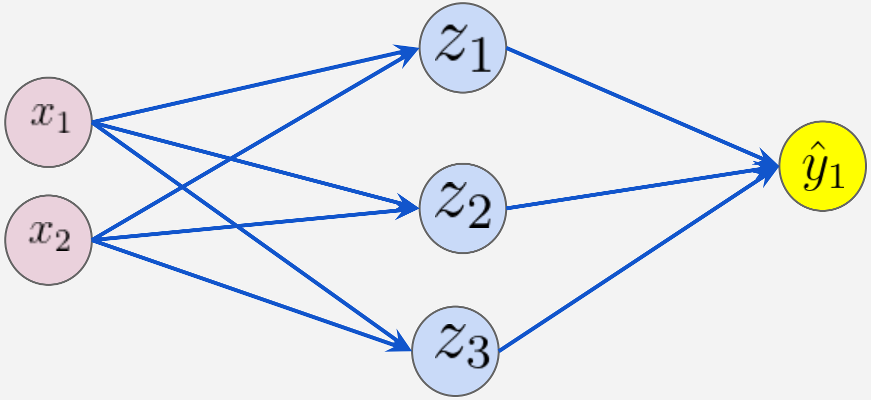



```
>>> y_true = [[0., 1.], [0., 0.]]
>>> y_pred = [[0.6, 0.4], [0.4, 0.6]]
>>> # Using 'auto'/'sum_over_batch_size' reduction type.
>>> bce = tf.keras.losses.BinaryCrossentropy()
>>> bce(y_true, y_pred).numpy()
0.815
```

Means Squared Error Loss (MSE Loss)

MSE Loss can be used with regression models that output real numbers

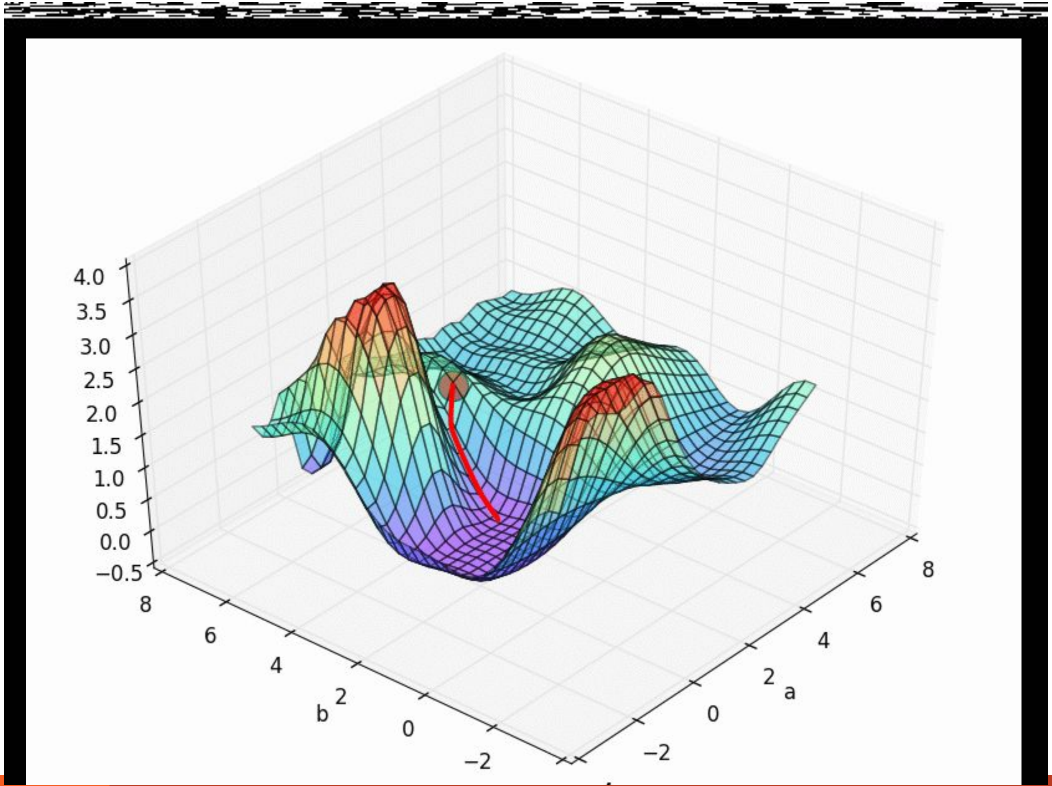
$$X = \begin{bmatrix} 4, & 2 \\ 2, & 1 \\ 5, & 8 \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$



$$\begin{array}{ccc} f(x) & & y \\ \begin{bmatrix} 30 \\ 80 \\ 85 \\ \vdots \\ \vdots \end{bmatrix} & \begin{array}{c} ✗ \\ ✗ \\ ✓ \\ \\ \\ \end{array} & \begin{bmatrix} 90 \\ 20 \\ 95 \\ \vdots \\ \vdots \end{bmatrix} \end{array}$$

$$J(W) = \frac{1}{n} \sum_{i=1}^{i=n} \Big(\underbrace{y^{(i)}}_{\text{Actual}} - \underbrace{f\Big((x^{(i)}; \mathbf{w}) \Big)}_{\text{Predicted}} \Big)^2$$

Gradient Descent Optimizer



$$W^* = \underset{\mathbf{W}}{\operatorname{argmin}}\, J(\mathbf{W})$$

1. Randomly pick initial values of wo and w1
2. Compute gradient $\frac{\partial J(\mathbf{W})}{\partial \mathbf{w}}$
3. Take small step in direction of gradient
4. Update wo and w1
5. Repeat 1-4 until convergence

Recap

1. History
2. Deep learning motivation
3. Perceptrons
4. Gradient Descent

Let’s play with Neural Nets

Check out the demonstration of Neural Nets at:

Tensorflow playground

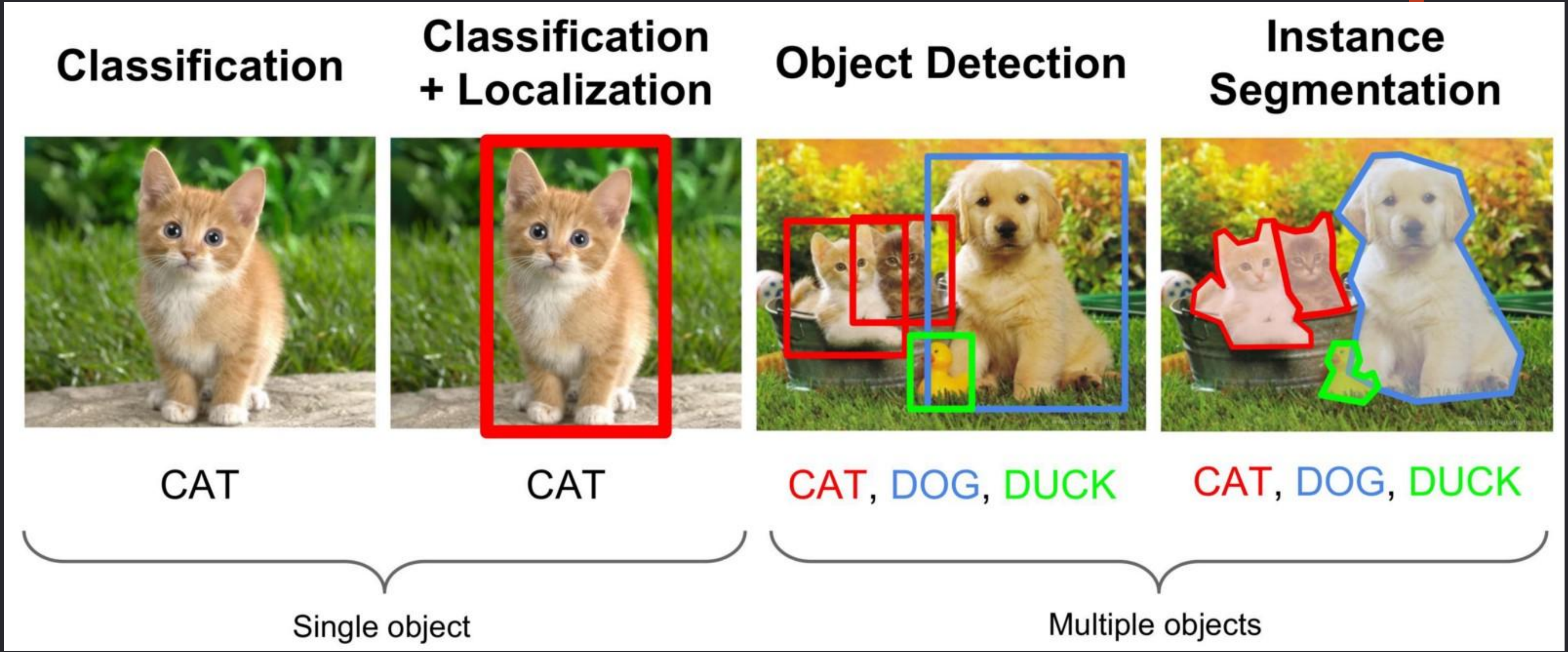
Optimization and Practical Details To Be Continued in Andreas' Talk:

Practical Deep Learning

Part 3: Deep Learning In the Wild

Convolutional Neural Networks

Common Problems

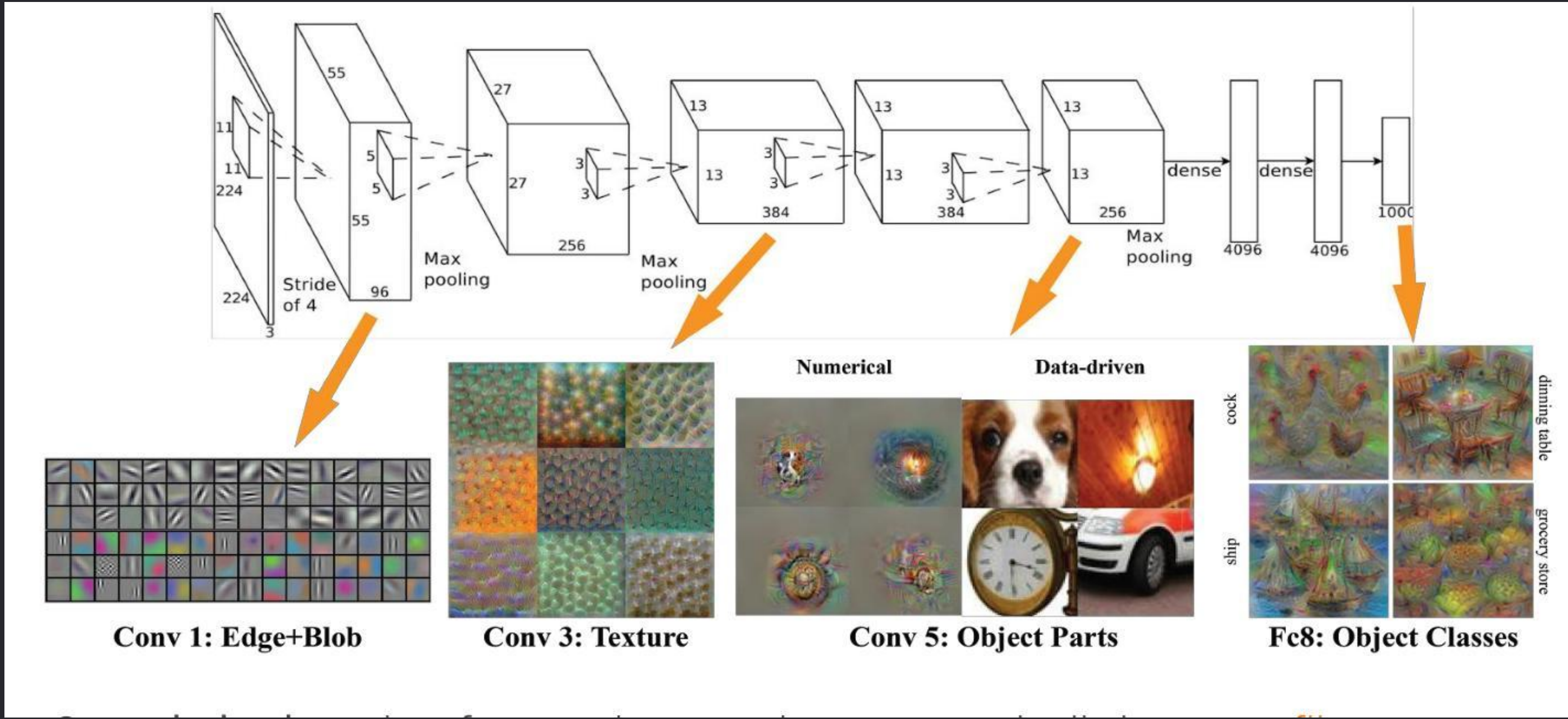


Convolutional Neural Network

Learn a complex representation of visual data using vast amounts of data

They are inspired by human visual cortex

Layers of a CNN



Layers of a CNN

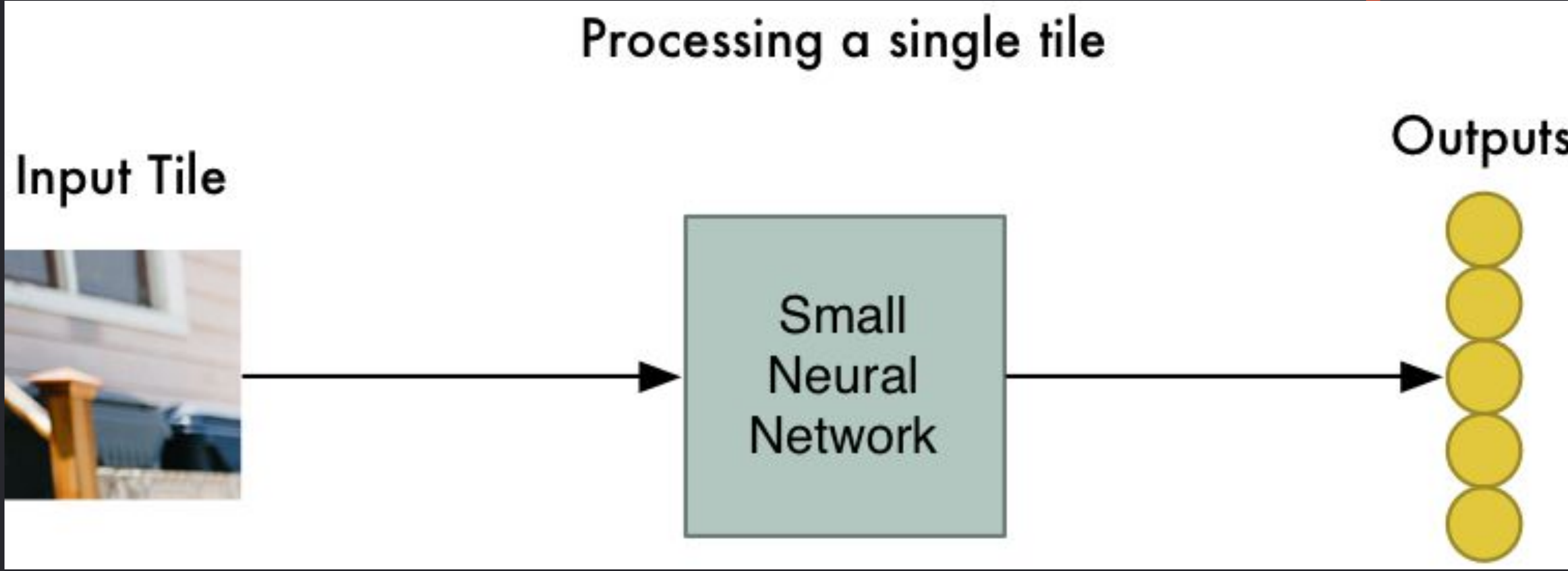
Convolution layer: A feature detector that learns to filter out not-needed information from an input using a convolutional kernel or (sliding window).

Pooling layer: Compute max or average value of a particular feature. Helps detect objects in some unusual places and reduces memory size

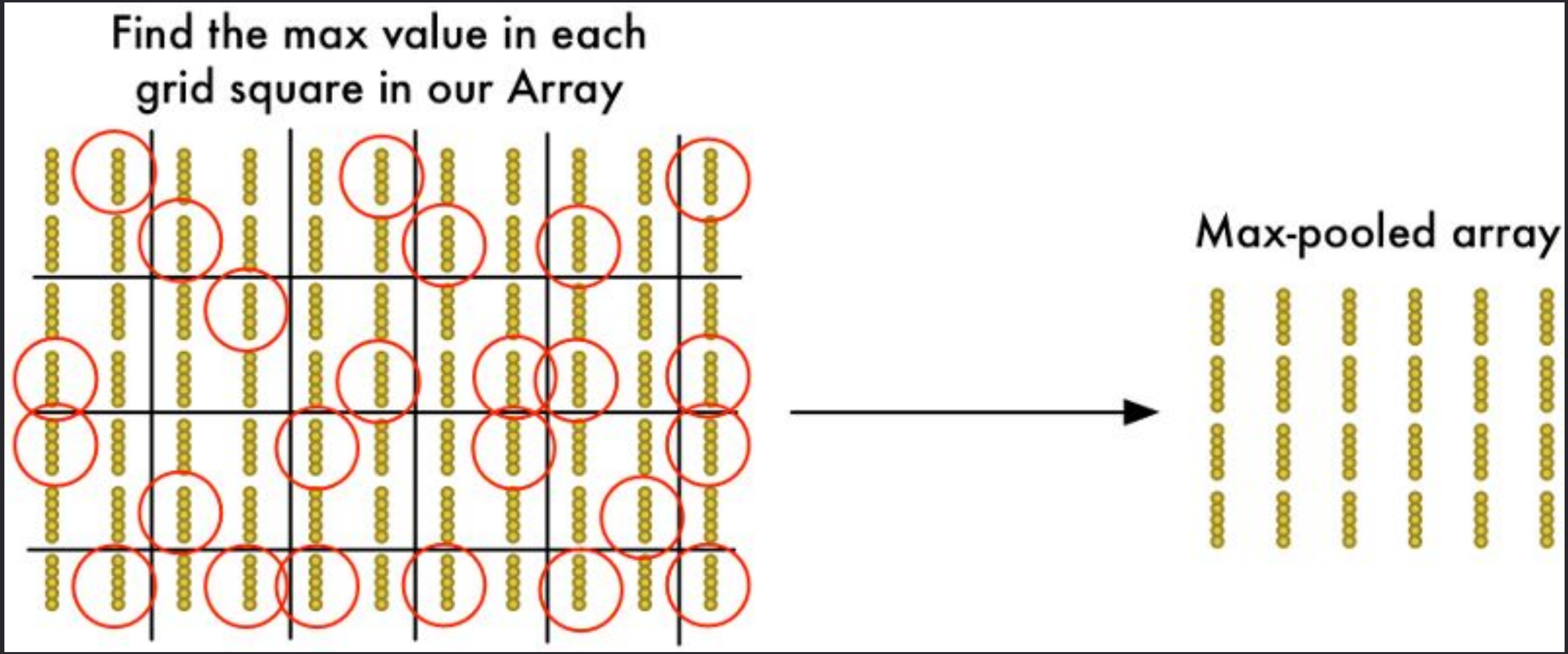
Consider an Input Image

Convolution Layer

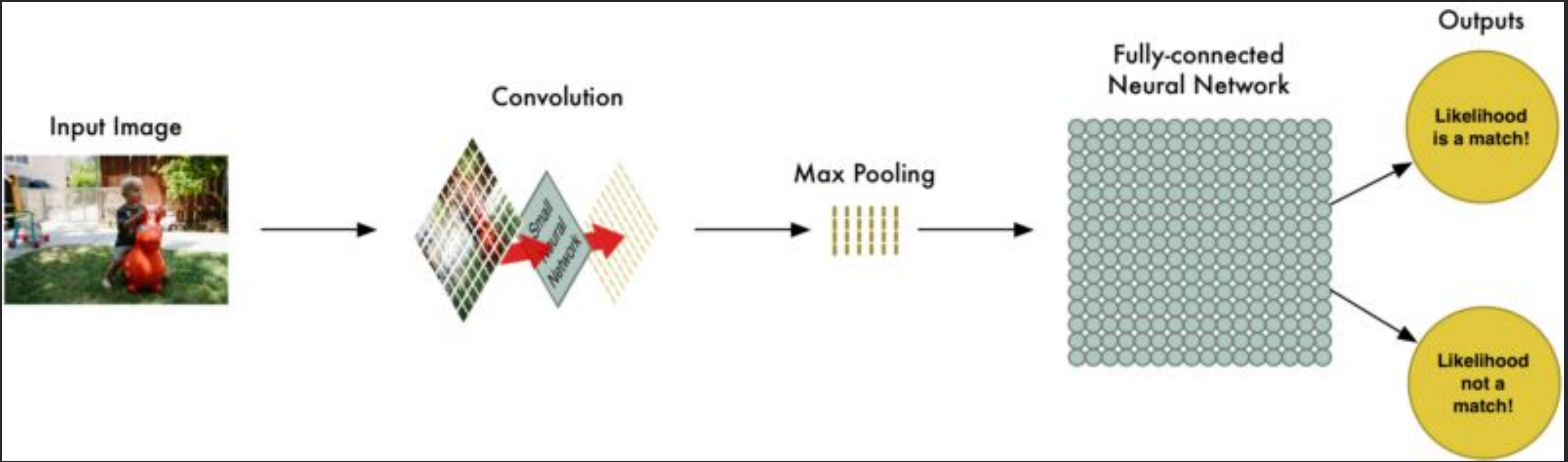
Pooling Layer



Pooling Layer

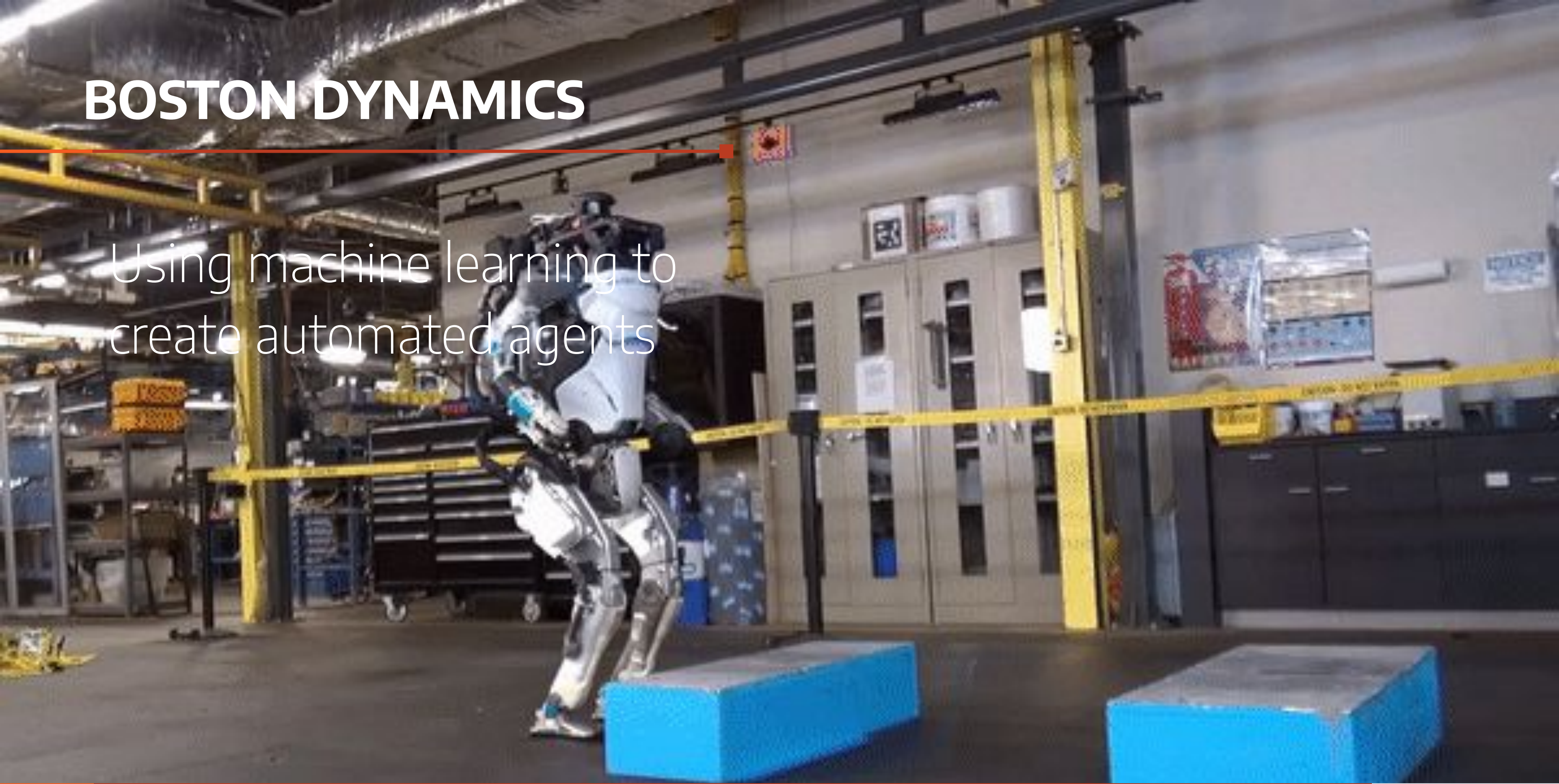


Complete Convolutional Neural Net

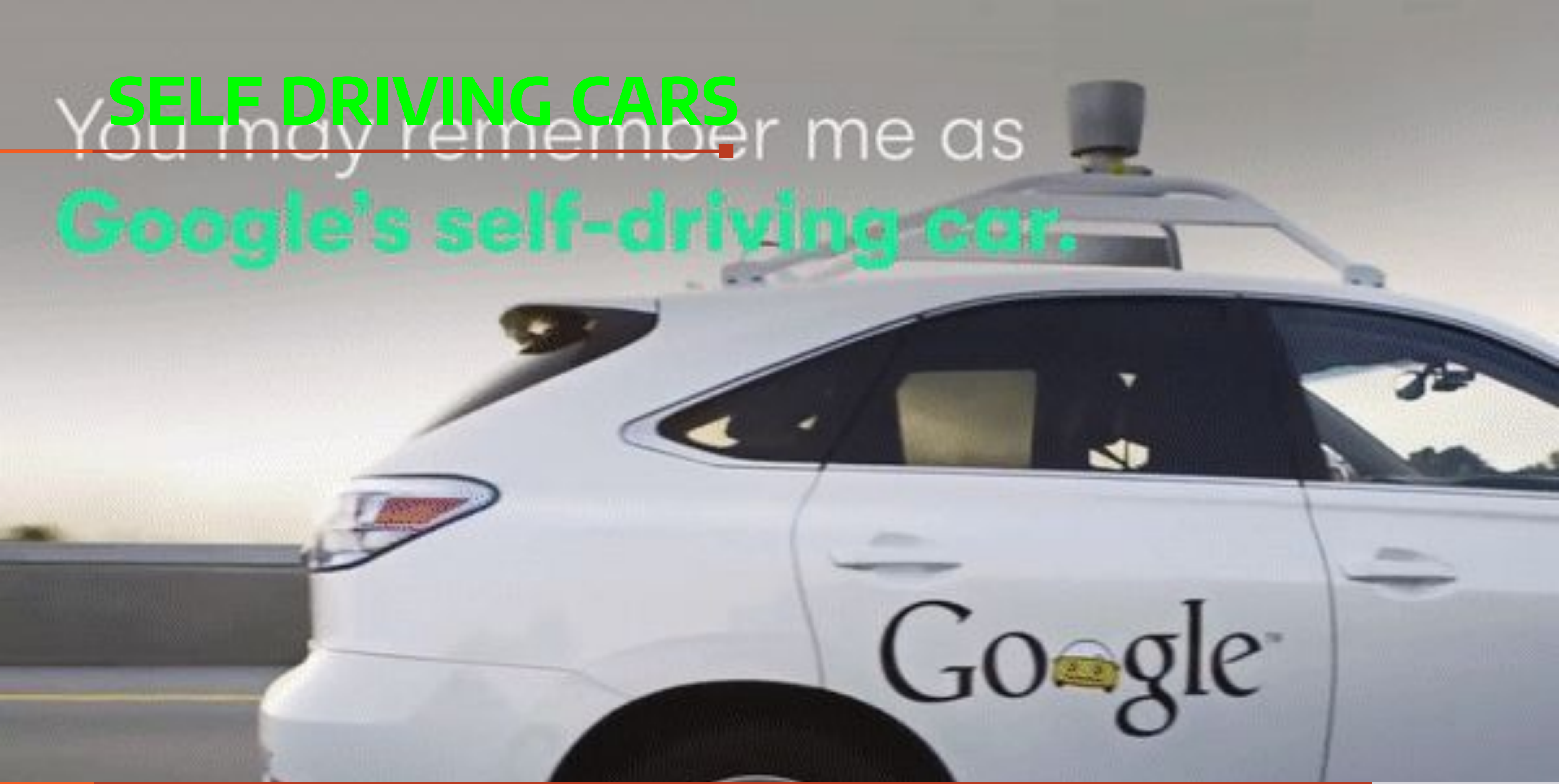


BOSTON DYNAMICS

Using machine learning to create automated agents



SELF DRIVING CARS



Wait... what about Africa?
Where is the Ai?

Ocula: Automated Mobile Microscopic Diagnosis

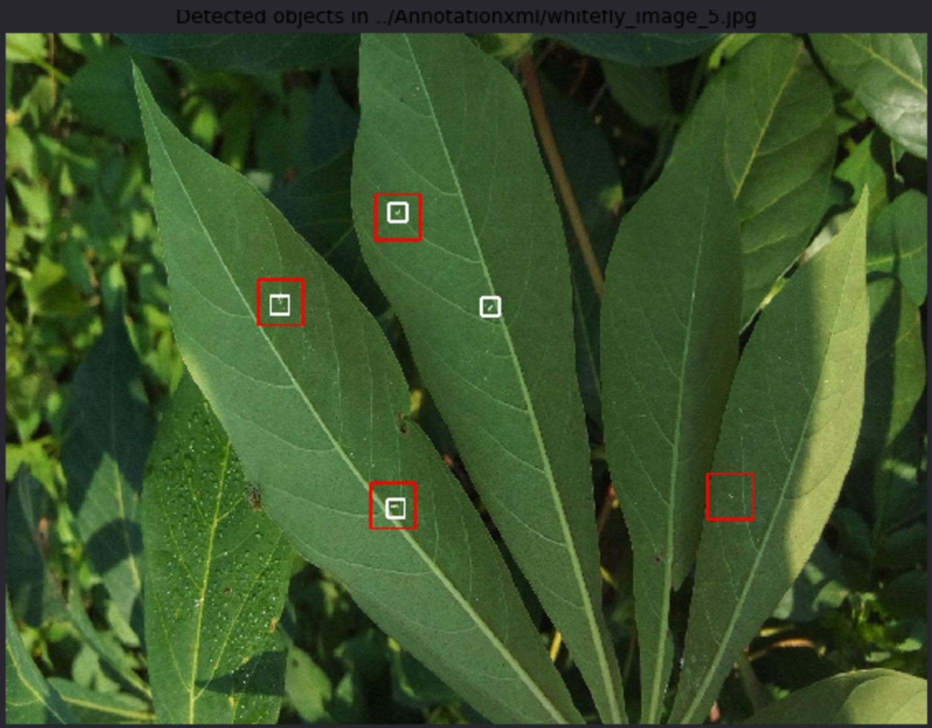
A platform for carrying out visual microscopic tests automatically, on a smartphone or netbook

AI RESEARCH

Whitefly Detection

In progress

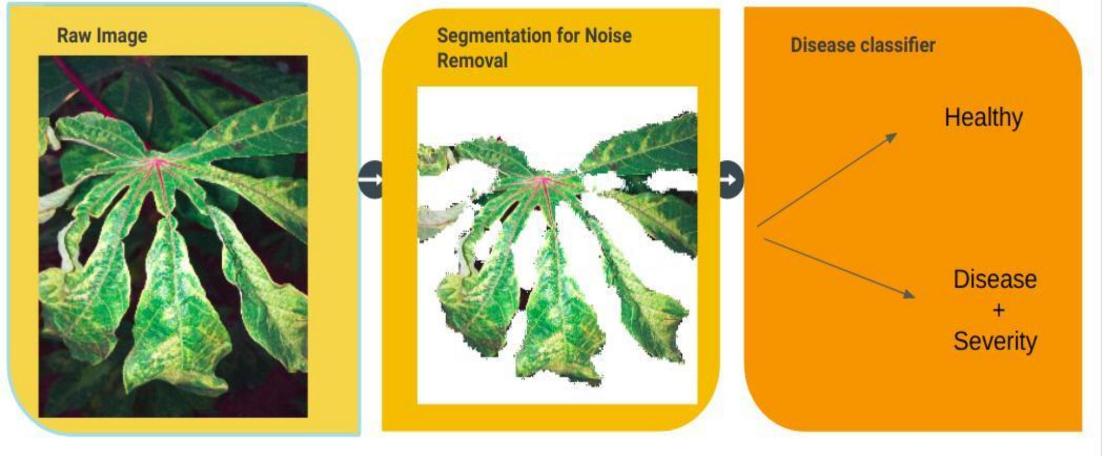
Using Object detection with convolutional Neural Networks to detect and count Whiteflies on Cassava leaves.



AI RESEARCH

Leaf Segmentation and Diagnosis

Using CNNs (Semantic Segmentation) to automatically remove background noise from leaves, to perform more accurate disease diagnosis.



Using Machine Learning To Analyse Radio Content In Uganda

PULSE LAB KAMPALA

The aim of the studies was to understand what type of information can be obtained from radio talk in Uganda and how it might be useful to advance and inform sustainable development and humanitarian action.

Part 5: Deep Learning For Real

In Python with Numpy and Tensorflow

Starts at: 4:10 PM (Kampala time)

Link to Material: shorturl.at/lCM04

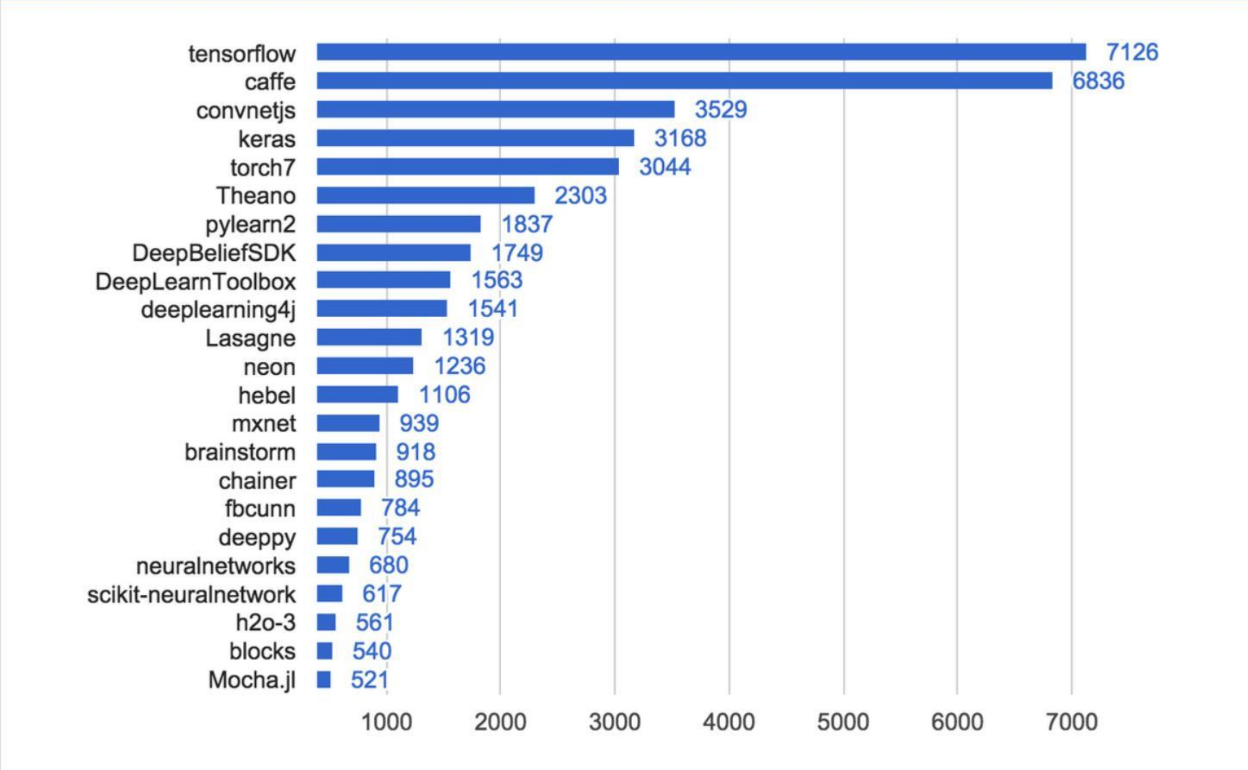
How to get started: Requirements

Large data set with good quality *(input-output mappings)*

Measurable and describable goals *(define the cost)*

Enough computing power *(AWS GPU Instance)*

Deep Learning tools: It's all open source



The most supported language by most libraries is **Python.**

Other languages are also supported: Java, C, JavaScript, Lua, R and Rubi.

Take aways

Machines that learn to represent the world from experience.

We haven't figured out creativity and human-empathy.

Deep Learning is no magic! Just statistics in a black box, but exceptional effective at learning patterns.

Transitioning from research to consumer products. Will make the tools you use every day work better, faster and smarter.

Ethical Considerations

- Fairness Accountability Transparency
- Bias

“

Will machines think?

Thank You!

ANY QUESTIONS?