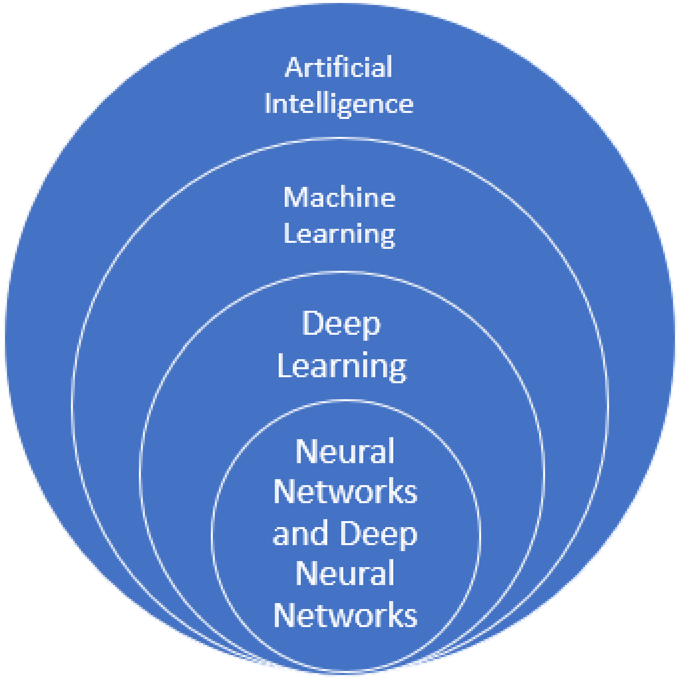
Faculty of Engineering and Technology
School of Computer Science and Mathematics

7144COMP Deep Learning Concepts and Techniques

Dr Paul Fergus, Dr Carl Chalmers
{p.fergus, c.chalmers}@ljmu.ac.uk
Room 714, 629 Byrom Street



Lecture 1

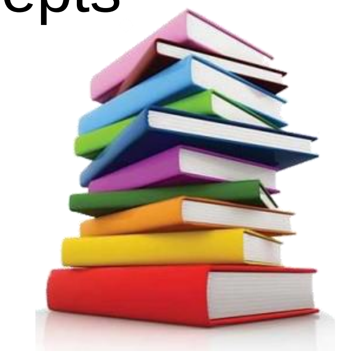
Module Overview and Introduction

In this session…

- We will cover:
  - – Module Overview
    - Aims & Assessment
    - Learning Activities & Learning Strategy
    - Achieving Success
    - Canvas Support
    - Recommended Reading
  - – Introduction to Deep Learning (DL)
    - Definition, use and emergence
    - Basic Concepts, Examples

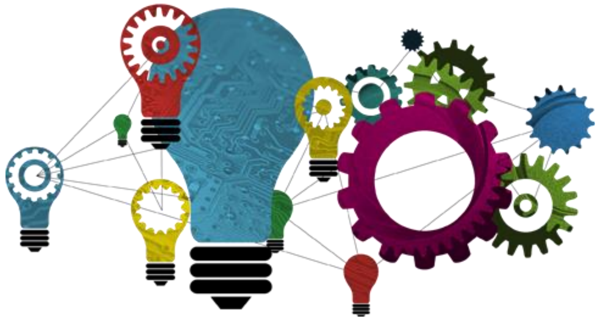
Learning outcomes

- At the end of this session you should understand:
  - The aims of the module
  - How you will be assessed
  - How you will learn on this module
  - How you can succeed on this module
  - The different forms of support available on this module
  - Have a basic understanding of Deep Learning and related concepts



Module Overview: Aims of the Module

- To develop knowledge of effective and academic understanding of deep learning at masters level and provide guidance on the purpose, design and development of deep learning projects.
- To provide an understanding of how the range of tools, techniques and algorithms can be most appropriately applied.
- To provide help on establishing best practice deep learning design and development principles to successfully complete a deep learning project.



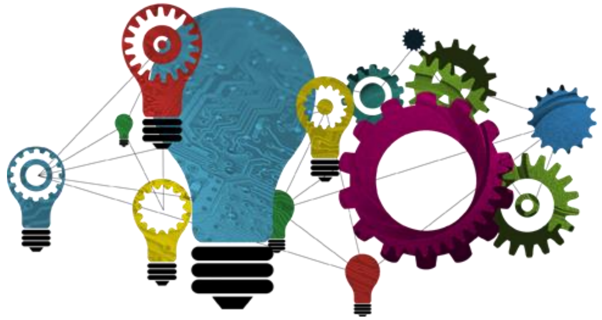
Module Overview: Assessment on the Module

- This module is assessed using:
  - Coursework 1, is worth 40% of the final module mark which is individual essay on the theoretical principles of deep learning
  - Coursework 2, is worth 60% of the final module mark which is the development of deep learning project
- The module has five learning outcomes:
  1. Demonstrate a critical understanding of the theoretical principles and objectives of Deep Learning (DL)
  2. Critically assess and select a range of DL concepts and techniques
  3. Critically select appropriate DL algorithms and architectures to solve particular tasks
  4. Implement and test different DL algorithms and architectures using Python and associated frameworks
  5. Evaluate DL algorithms and architectures to determine their strengths and weaknesses
- The learning outcomes are mapped to the assessment components as follows:

Report	Prototype
1	3
2	4
	5

Module Overview: Learning Activities

- Each week there is
  - 1 x 1 hour lecture followed by a 1 x 1 hour tutorial and a 1 x 1 hour lab
- The coursework is a **PRACTICAL** assessment that requires you to ***gain experience*** via:
  - The lab exercises
    - Applying what you have learned to new problem-based scenarios
  - The lecture and tutorial sessions
    - Observing the tutor



Module Overview: Learning Strategy

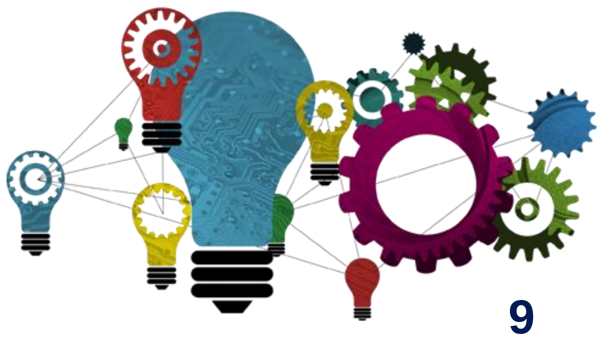
- Each week the theory and experience you gain will **build** upon the previous weeks

- So the module is **NOT** designed for you to dip in and out of

- It is designed for you to follow a sequence of learning experiences
  - If you miss a lecture or lab, YOU need to work on them in your own time before the next learning session
  - If you get stuck, simply ask for help at the next learning session
    - The ‘moral of story’ is ‘keep up so you don’t fall behind’

Module Overview: Learning Strategy

- There are 167 hours of private study time specified on the module specification
- In that time you should ‘play’ with the technologies in preparation for the assessment.
- You should also read through the designated reading material which is available on Canvas



Module Overview: Canvas Support

- Each week on Canvas you will find the following materials in the ‘Modules’ area:
  - Lecture slides
    - Bring your note books as more will typically be discussed than is provided in the slides
  - Lab exercises
    - The lab exercises can be found on your dedicated lab computer
  - Web links and further reading
    - Links to websites and further reading that will support you in your learning
- There will be a checkpoint assessment on Canvas (week 5)
- This quiz will assess how well you have learned the concepts covered to date

Module Overview: Canvas Support

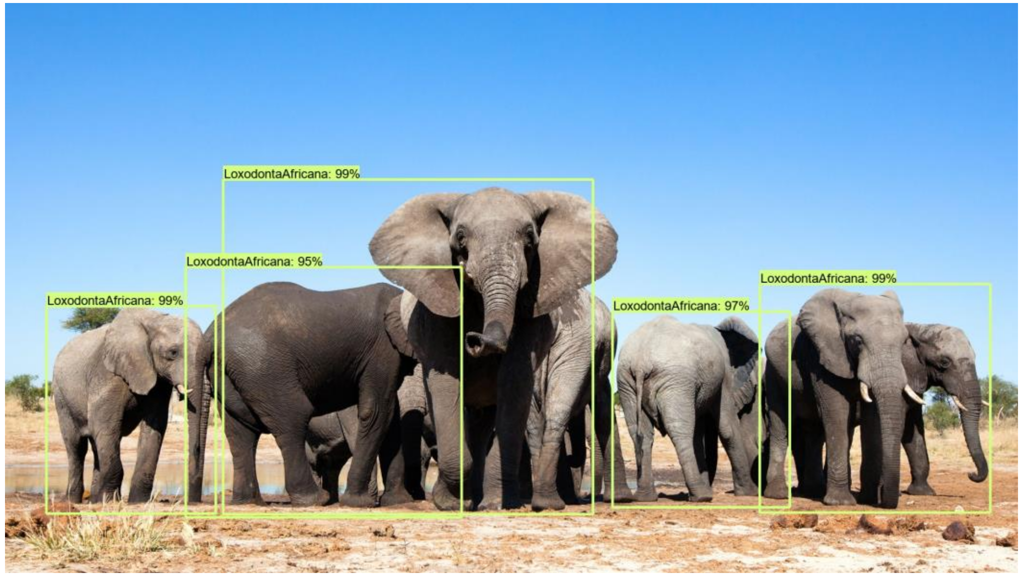
- When we start to learn the TensorFlow Object Detection API you will also find a set of additional resources in the ’Modules’ area to help you with your learning
- In the ‘Modules’ area you will also find:
  - Module Proforma
    - The ‘specification’ for the module
  - Module Handbook
    - An informative ‘design’ of the module
  - Reading List
    - Access to books available in the library and some as eBooks
- ‘Working outside the lab’ Area
  - You can remote into your designated lab computer using the instructions provided
  - Although all of the module software is open source it computationally expensive so do as much as you can on the lab computer

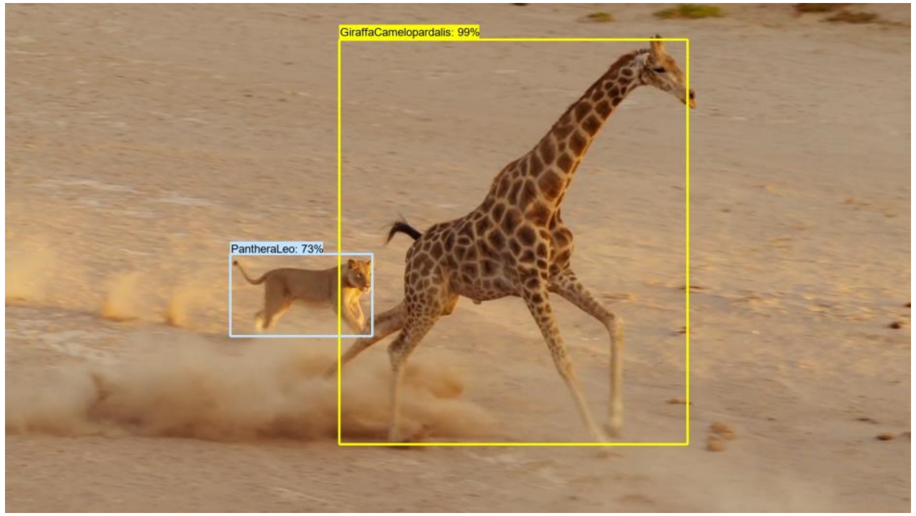
Module Overview: Software

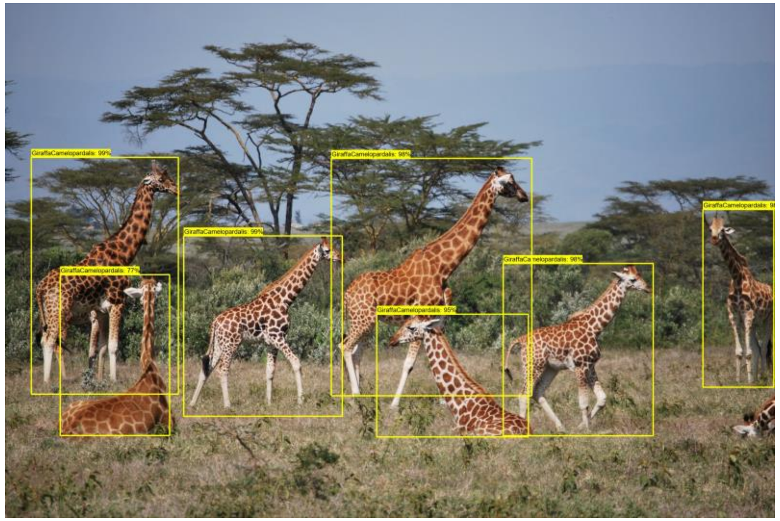
- We will use a variety of big data tools and packages in this module
- Python is the default programming language and is used throughout the module

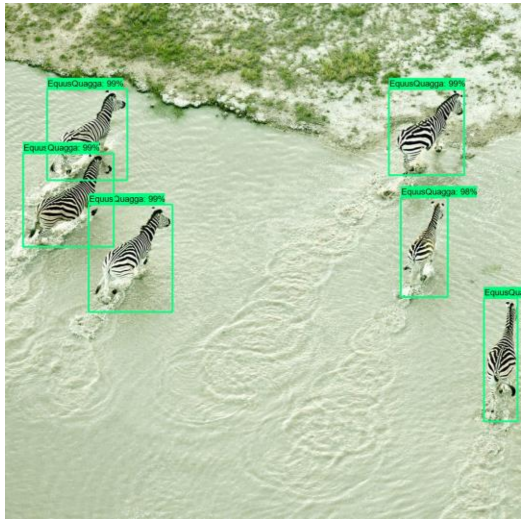
Module Overview: Labs

- Python and Juypter Notebook to write and execute your code
- Building MLPs using TensorFlow and Keras
- Tagging and data pre-processing for object detection
- TensorFlow Object Detection API 2.0
- Saving and inferencing our trained models









Module Overview: Lecture Structure

Lecture 2: Artificial Neural Networks

- ANN's and their biological comparison
- Perceptions
- Neural Networks (MLP)
- ANN types and uses (CNN'S, Feed forward etc.)
- Activation Functions
- Multi-Class Classification Considerations

Lecture 3: Optimisation (Hyperparameter Tuning):

- Loss Functions
- Optimisation Algorithms (SGD, ADAM etc.)
- Local and Global Minima, local Maxima, Global Maxima
- Learning Rate
- Configuring Hyperparameters

Lecture 4: Backpropagation

- Introduction to backpropagation
- Chain rule
- Computational graphs
- The vanishing gradient
- Overcoming the vanishing gradient issue
- Weight Initialisation
- Regularisation

Lecture 5: Introduction to Convolutional Neural Networks (CNNS)

- Introduction and biological comparison
- Hubel and Wiesel
- Image Data
- Image Filters and Kernels
- Convolutional Layers
- Pooling Layers
- Transfer Learning

Module Overview: Lecture Structure

Lecture 6: Image Classification and Object Detection Part 1

- Computer vision overview
- Historical context
- Why computer vision is useful
- Data-Driven Approach (Collect a dataset of images and labels)
- Famous datasets
- Hardware Accelerated Deep Learning (CPU vs GPU)
- Training and Associated Hardware (Development Systems, Training Systems, Inferencing Systems)
- Tensor Processing Unit (TPU)
- Other Hardware Considerations
- Distributed Training
- Model Parallelism

Lecture 8: Model Architectures Part 1:

- Introduction
- Progression of Model Architectures
- Single Shot Detector (SSD) MobileNet
- SSD Benefits and Limitations
- YOLO Family
- YOLO Benefits and Limitations
- R-CNN
- R-CNN Benefits and Limitations
- Fast –RCNN
- Fast – RCNN Benefits and Limitations
- Faster-RCNN
- Faster – RCNN Benefits and Limitations

Lecture 7: Image Classification and Object Detection Part 2

- Object Recognition
- Image Classification
- Object Detection
- Semantic Segmentation
- Object Segmentation
- Skip connections
- Data Augmentation (Horizontal Flips, Random Crops and Scales, Colour Jitter)

Lecture 9: Model Architectures Part 2:

- EfficentNet
- EfficentNet Benefits and Limitations
- Summary and Key findings
- Frameworks
- TensorFlow Object Detection API

Lecture 10: Evaluation Metrics

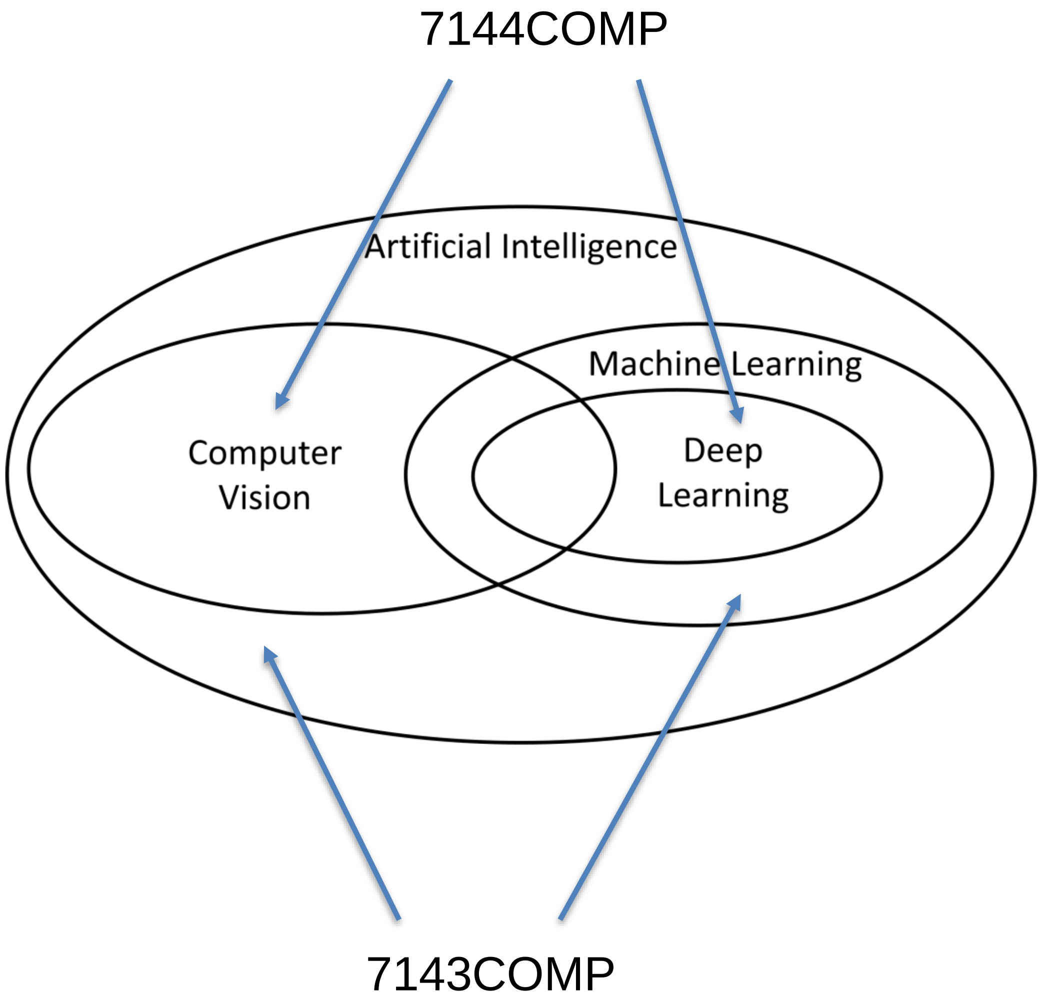
- Loss
- Intersection over Union
- Mean Average Precision
- Tensorboard

Lecture 11: Training Strategy and Summary

- Early Stopping
- Model Ensembles
- Large-Batch Training: Scale Learning Rates
- Summary and Open Problems

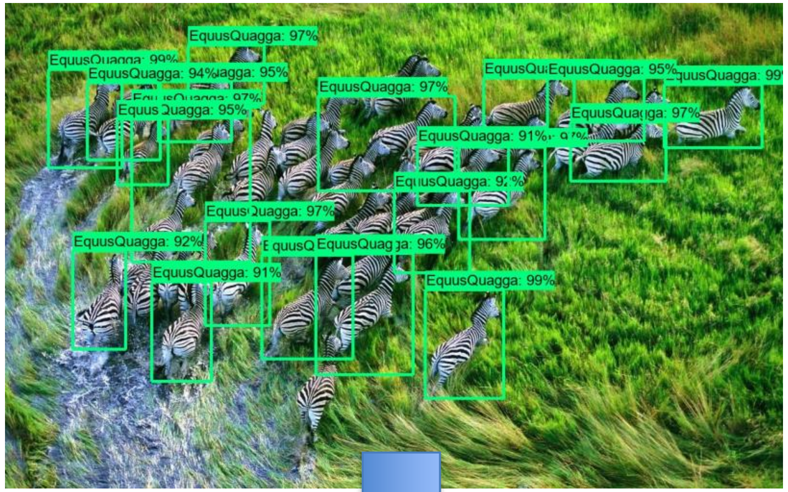
Introduction to Deep Learning

- Just like Machine Learning, Deep Learning (DL) is a subset of AI which uses biologically inspired networks to solve a given problem
- Neural networks which are a key component in most DL architectures are modelled on the human brain which allow computers to learn in a similar manner
- Some of common problems currently solved using DL includes pattern recognition, time series predictions, signal processing, anomaly detection, control and automation and computer vision



Introduction to Deep Learning

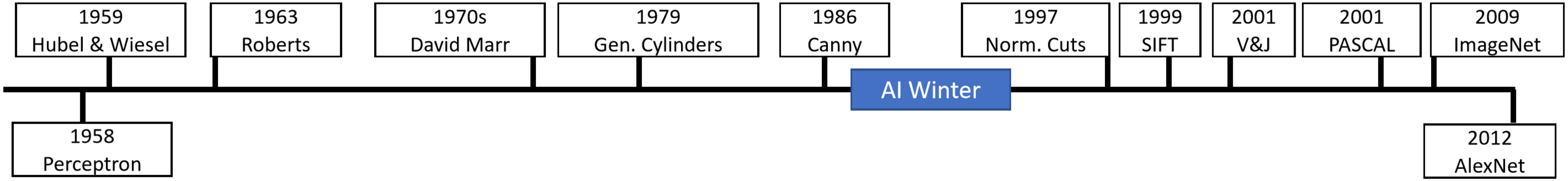
- Like with traditional ML you will find that most of the common tools and techniques are also applicable in the DL domain
- You still need to undertake elements of the Exploratory Data Analysis (EDA) process, partition your data in train, validation and test sets and evaluate your model using the appropriate metrics
- However, many of the similarities end here as the fundamental architecture of DL models differ from their traditional ML counter parts
- One of the key differences is automated feature extraction whereby the network itself extracts feature it deems to be important



What features would you use if you had to do this manually?

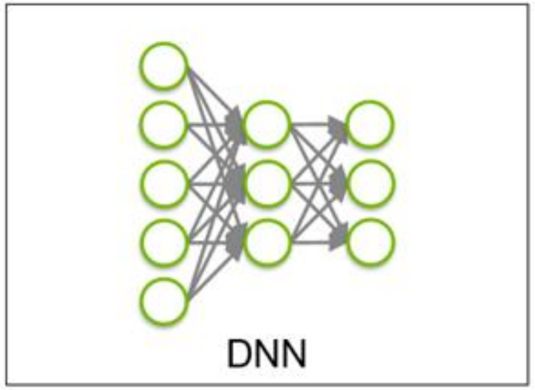
Introduction to Deep Learning

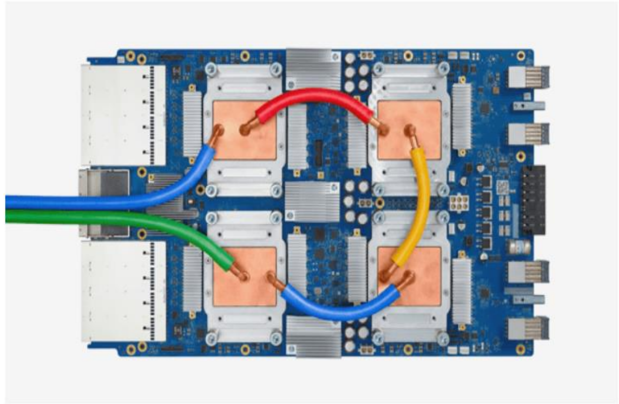
- Automatic feature extraction is an important advancement in DL that has simplified the training pipeline, removed in the need for domain knowledge and improves the quality of features extracted
- There has been and explosion of DL in both academia and industry in recent years sparking what is commonly known as the AI summer
- In 2012 deep learning started to become mainstream when Krizhevsky, Sutskever, and Hinton published their work on AlexNet
- They managed to bring decade of work together for the first time



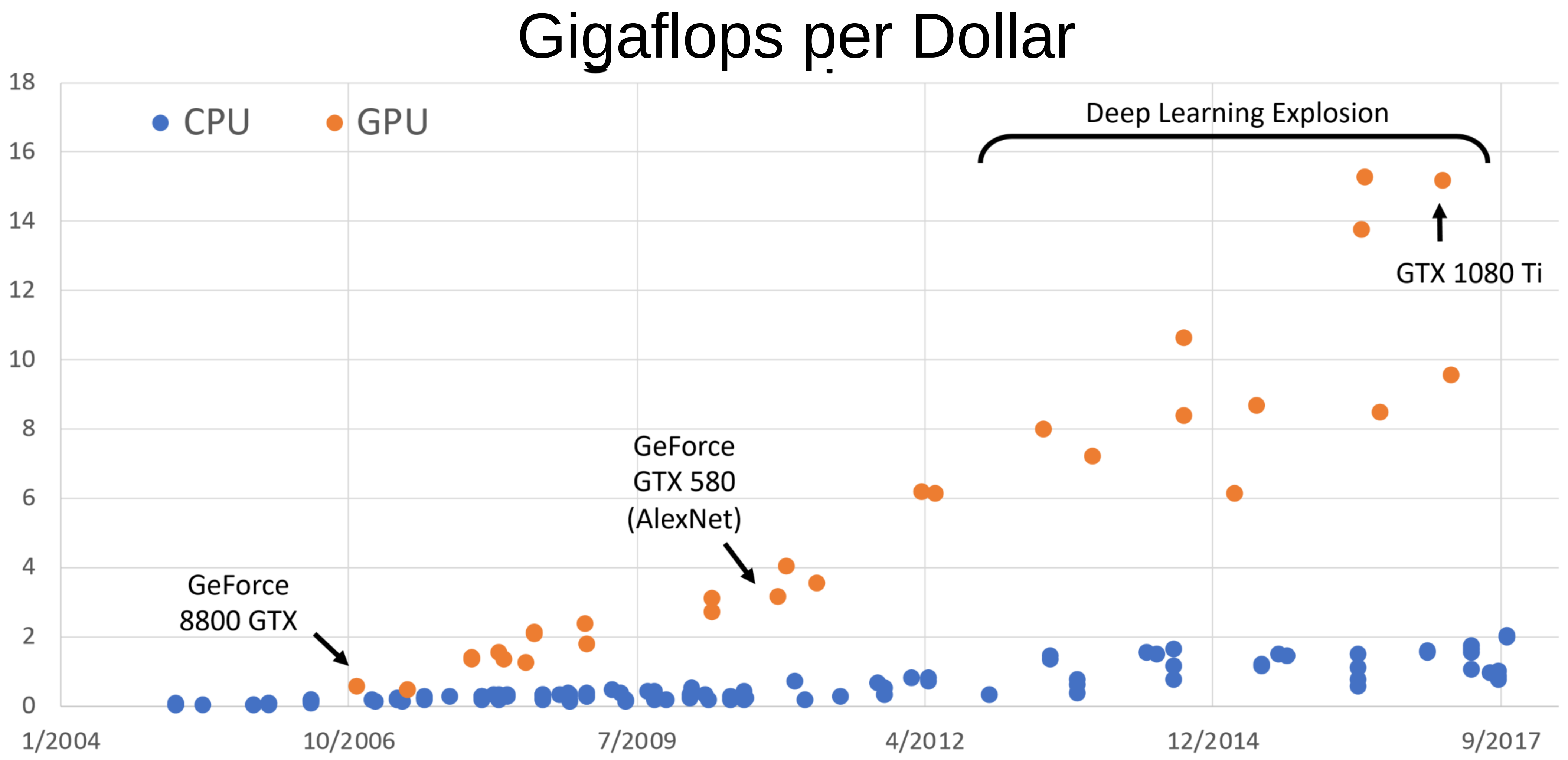
Enablers of Deep Learning

- Although DL in not new and can be traced back to the early 70's there are three fundamental components that have seen significant advancement help to drive the widespread adoption of DL
- These are advancements in deep neural network architectures, Graphics Processing Units (GPU's), Tensor Processing Units (TPU's) and the availability of large and ever-growing high-quality datasets





Enablers of Deep Learning

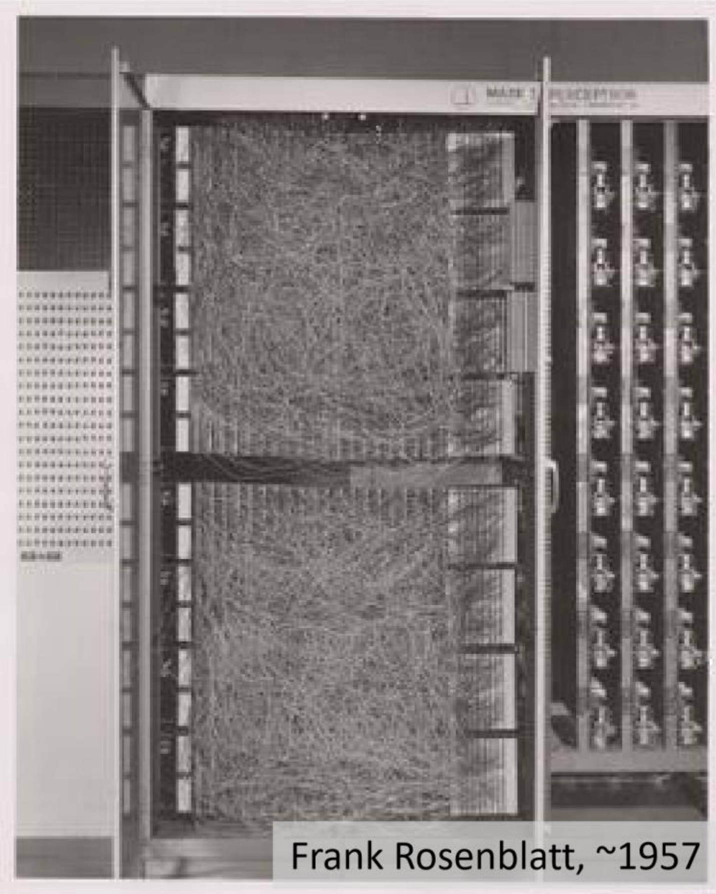


Enablers of Deep Learning

- The combination of these technologies are driving some of the most impressive advancements in innovative solutions such driverless cars, smart cities and smart devices such as Alexa
- Although you may not realise it DL is already playing a fundamental role our daily lives to help, support and enhance the things we do
- For example, recommender systems, healthcare, safety, finance, entertainment and multitude of services in many of the devices we own

Enablers of Deep Learning

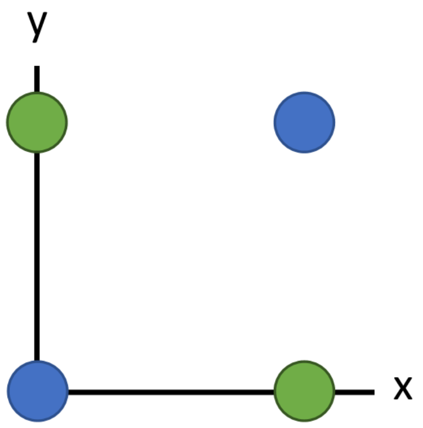
- Perceptron (discussed in detail in the next lecture) one of the earliest algorithms that could learn from data
- Implemented in hardware! Weights stored in potentiometers, updated with electric motors during learning
- Connected to a camera that used 20x20 cadmium sulfide photocells to make a 400-pixel image
- Could learn to recognize letters of the alphabet


Frank Rosenblatt, ~1957

Enablers of Deep Learning

- Today we would recognise a perceptron as a linear classifier
- The invention of a perceptron got people excited as it was a mechanism to allow a machine to learn novel things by using data without using explicit programming
- However Minsky and Papert,1969 showed that perceptron's could not learn the XOR function (we will see this is lab 1)
- This is addressed using an MLP

X	Y	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0



Enablers of Deep Learning

- Rumelhart, Hinton, and Williams, 1986 (Back propagation)

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial w_{ji}}$$

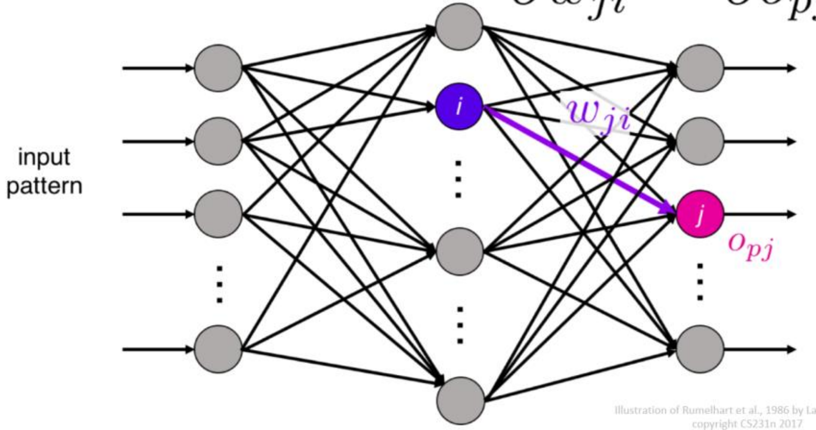
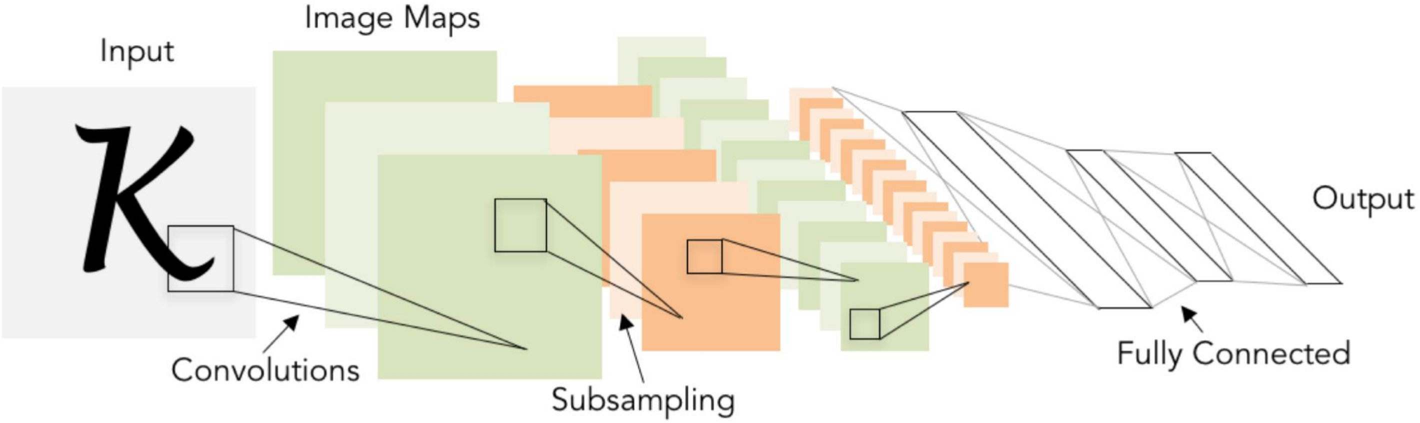


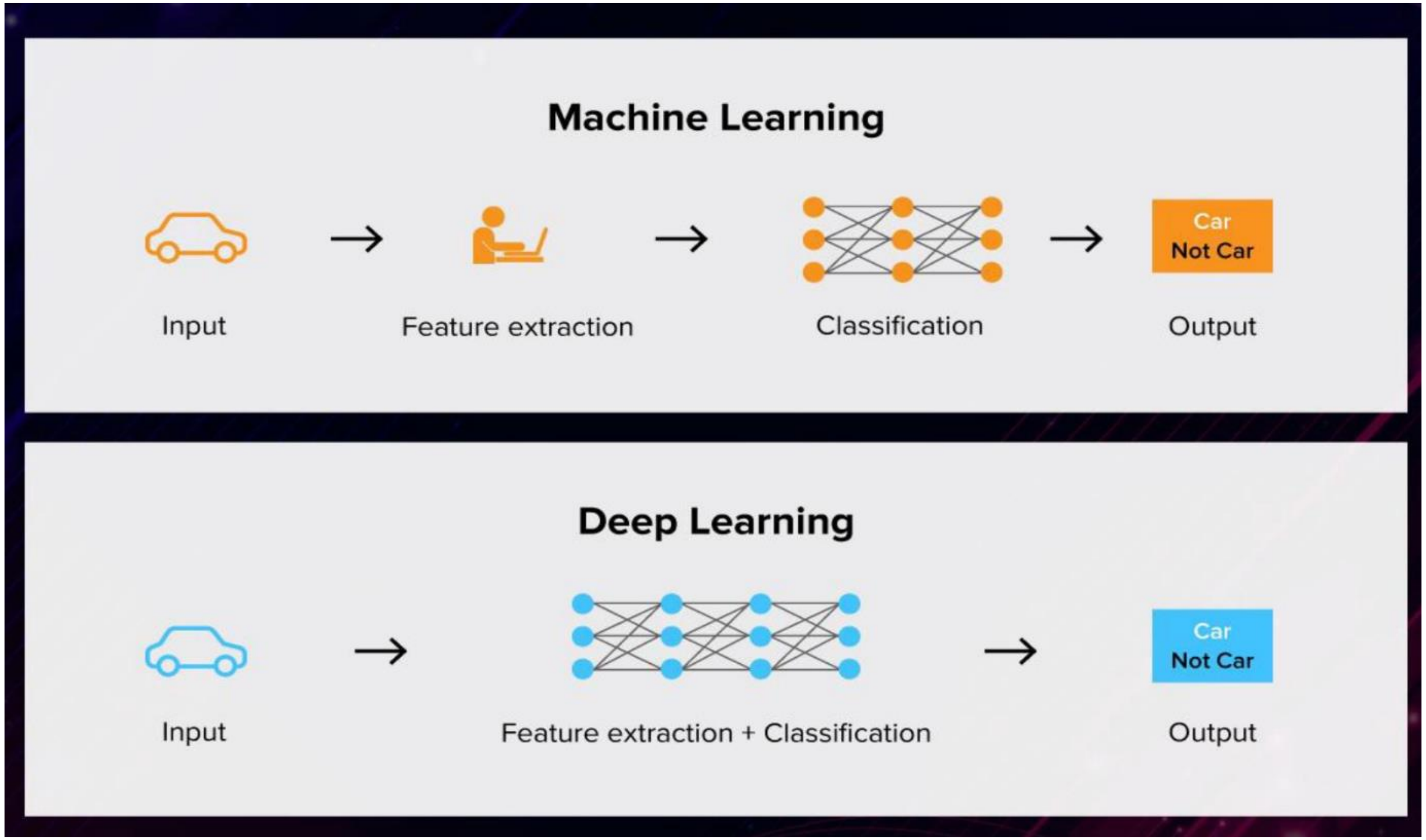
Illustration of Rumelhart et al., 1986 by Lane McIntosh, copyright CS231n 2017



So what's the difference between DL and ML?

- ML and DL are very closely related in the sense that they allow machines to learn from prior data
- The term ML is a catch all for any machine that learns in this way. Whereas DL is a specific set of methods and techniques to enable a machine to learn and make decisions using very deep and complex networks
- However perhaps one of its most notable differences is for its ability to replace the human driven feature extraction process and incorporate this this stage within the networks itself to automatically decided which features best describe the data

So what's the difference between DL and ML?



So what's the difference between DL and ML?

- In certain applications such natural language processing and computer vision it is very difficult to apply algorithms such as SVM’s and Random Forests to understand and translate language or detect and classify objects within images
- However, these are tasks that DL algorithms find relatively easy to solve
- Therefore, it is always important to understand what ML and DL algorithms can and cannot do before selecting an appropriate model to use
- In cases where an ML algorithm, such as SVM or RF, would suffice they should always be selected (where there is also time and sufficient expertise to extract the required features from the data) for three primary reasons
- Firstly, computational efficiency, second model interpretation, and third the amount of data you have

So what's the difference between DL and ML?

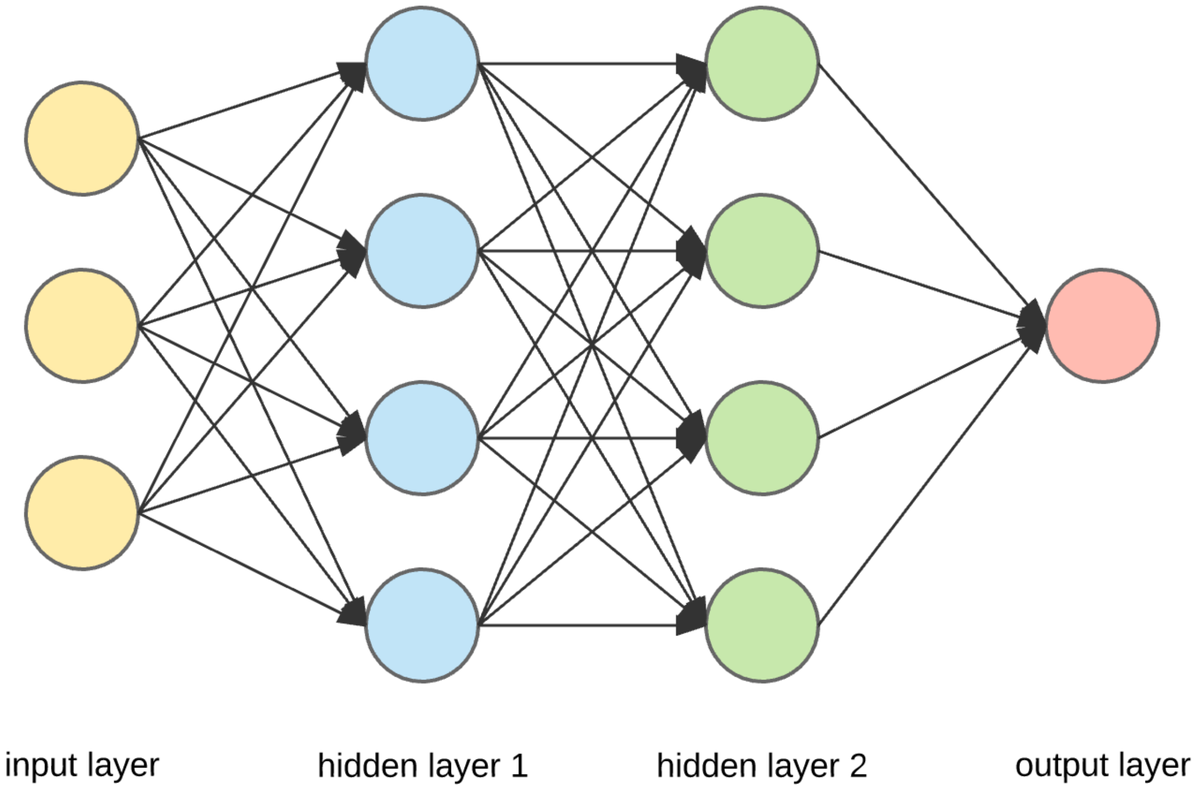
- Although, DL models have proven to be able to solve some of the most challenging of problems, they can be data hungry and very computationally expensive
- Careful consideration of the hardware requirements for training and hosting complex DL models is required before committing to a DL based solution
- For example, building an application to detect 29 of the most common species found in Sub-Sahara Africa using HD camera trap images would take a single compute node with 4x NVidia Quadro M4000 GPU’s roughly four days to train a model with 1000 images per species

So what's the difference between DL and ML?

- In reality you would need a vastly larger dataset and additional hardware to support this such as a DGX1 or DGX2 which at time of writing rages between £100,000 and £300,000. Note that the DGX2 has 16x Tesla V100 GPU’s

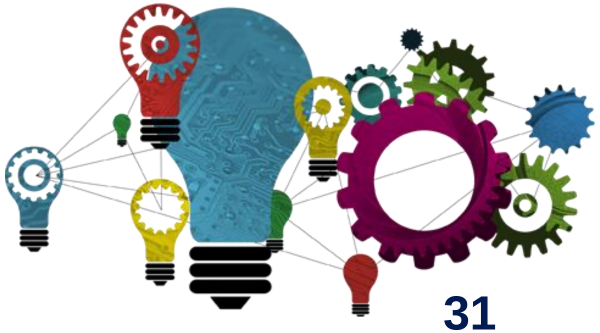
Introduction to Deep Learning

- DL employs algorithms to process data and imitate the thinking process or to develop abstractions
- With DL information is passed through each layer with the output of the previous layer providing inputs for the next layer
- The first layer in a network is call an input layer while the last layer is called the output layer
- All the layers between the two are referred to as hidden layer



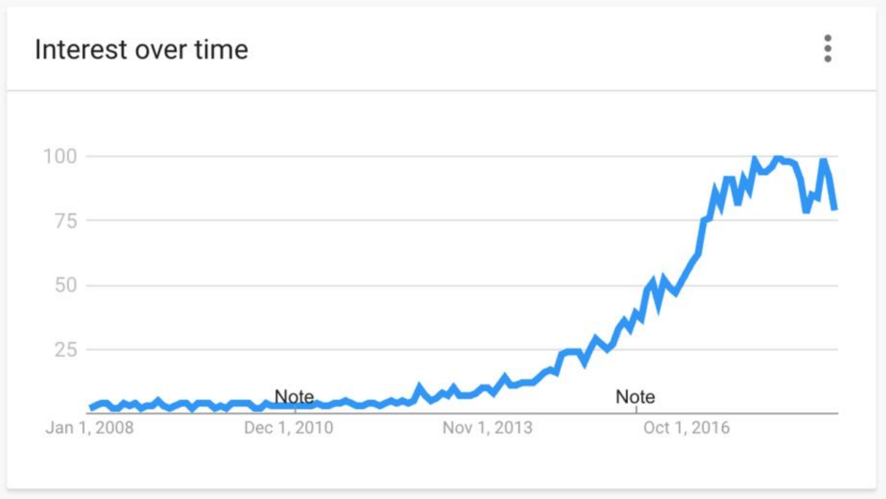
History of Deep Learning

- The history of deep learning can be traced all the way back to 1943 when the first computer model based on the neural networks of the human brain was created
- Since then DL has steadily evolved with only two significant breaks in its development both of which are known as AI winters
- One of the most significant breakthroughs in ANN was the development of the back-propagation algorithm which was applied to ANN in 1985 with a full implementation in 1989 at Bell Labs
- Backpropagation was combined with Convolutional Neural Networks in 1989 which was trained to recognised had written digits

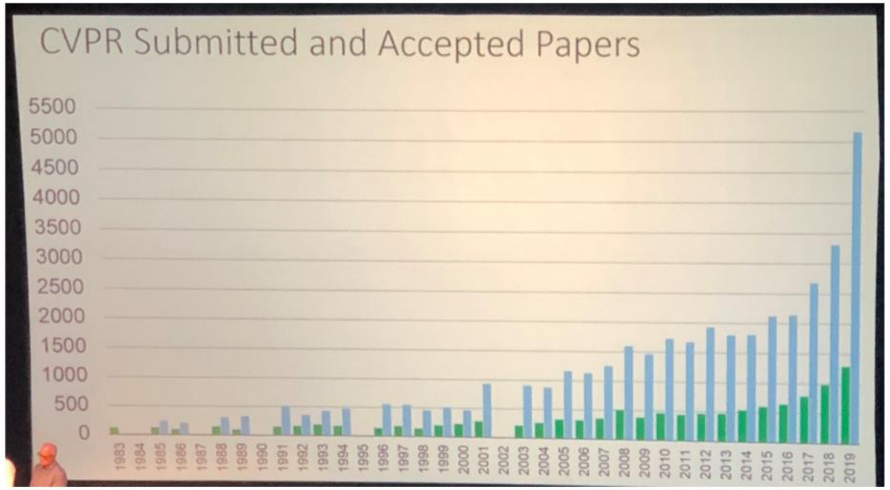


History of Deep Learning

- In 1999 the integration of GPU technology with DL marked the next significant breakthrough which increased computation speeds 1000% over 10-year period
- In 2009 a database containing 14 million labelled images was released which was an important milestone for image processing architectures interested in analysing images using ML techniques as ANN required large volumes of labelled images for successful training



Google Trends: “Deep Learning”



Publications at top Computer Vision conference

History of Deep Learning

- By 2011 advancements in GPU's meant it was possible to train CNN's without the layer by layer pre-training
- Since then the area of DL has expanded into many domains which focus on adjustments to neural network architectures, tools and techniques to speed up and improve the accuracy of models
- AI has significant fallout over the years where it promised a lot and delivered very little
- DL has changed this, and the results produces have proved to be very important as it is able to achieve meaningful and useful accuracy in many real-world applications

The success of Deep Learning

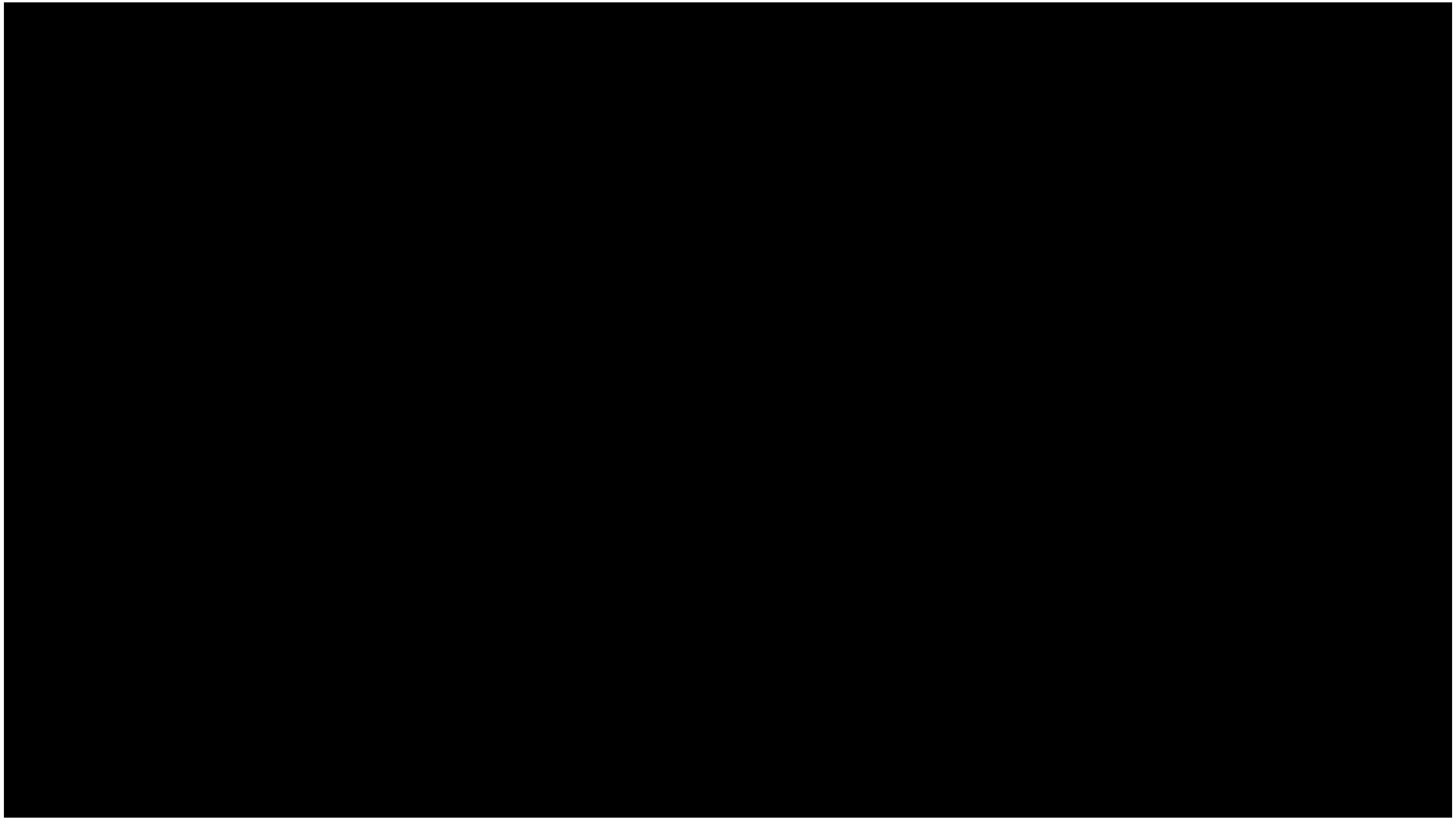
- The explosion of DL successes has been attributed to the supporting frameworks provided by the biggest global technology companies such as Google, Facebook and Microsoft
- These companies are at the forefront of applied AI and its adoption within many business sectors
- They have developed and offer DL frameworks that are free to use which have significantly eased to training and hosting of models

The success of Deep Learning

The success of Deep Learning

- DL has been used to outperform humans in highly intellectual and skilful games such as chess and go
- It has now become extremely difficult if not impossible to beet DL systems such AlphaGo
- Famous match between Lee Sedol (who was the winner of 18 world titles) and AlphaGo where AlphaGo won 4 – 1

The success of Deep Learning



Considerations

- Despite all of the success and achievements deep learning and computer vision still has some ways to go
- DL cant solve all of the problems and we need a joined up eco-system of technologies (symbolic and non-symbolic AI)
- Being able to recognise an object is one thing but being able to understand it is another

Next Session

- ANN’s and their biological comparison
- Perceptions
- Neural Networks (MLP)
- ANN types and uses (CNN’S, Feed forward etc.)
- Activation Functions
- Multi-Class Classification Considerations