Deep Learning with R

Poo Kuan Hoong

https://www.facebook.com/rusergroupmalaysia/

https://www.meetup.com/MY-RUserGroup/

Disclaimer: The views and opinions expressed in this slides are those of the author and do not necessarily reflect the official policy or position of ASEAN Data Analytics Exchange (ADAX). Examples of analysis performed within this slides are only examples. They should not be utilized in real-world analytic products as they are based only on very limited and dated open source information. Assumptions made within the analysis are not reflective of the position of ADAX.

Agenda

- Introduction
  - Where are we now…
  - What is Deep Learning?
- Human Brain and Neural Networks
- Deep Learning: Strengths and Weaknesses
- Deep Learning Libraries
  - TensorFlow
  - MXNET
- Demo

About me

Poo Kuan Hoong, http://www.linkedin.com/in/kuanhoong

ADAX

- Senior Data Scientist

nielsen

- Senior Manager Data Science

MMU
MULTIMEDIA UNIVERSITY

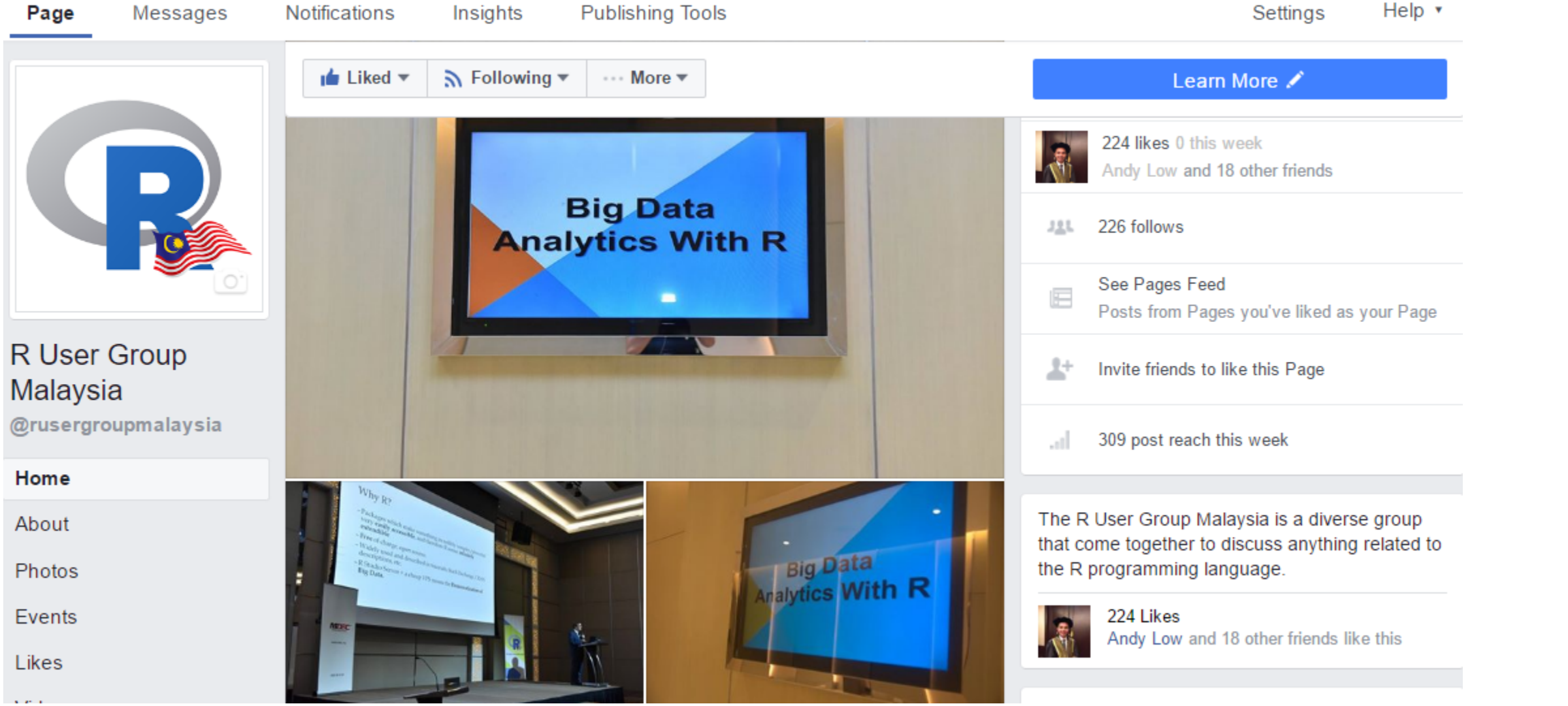
- Senior Lecturer
- Chairperson Data Science Institute

- Founder R User Group & TensorFlow User Group
- Speaker/Trainer

Malaysia R User Group

The Malaysia R User group's aims are:

- To provide members ranging from beginners to R professionals and experts to share and learn about R programming
- A knowledge sharing and presentations platform in relation to the techniques using R.



https://www.facebook.com/rusergroupmalaysia/

Malaysia R User Group - (myRUG)

Home | Members | Sponsors | Photos | Pages | Discussions | More | Group tools | My profile

Change photo

Kuala Lumpur, Malaysia
Founded Jun 5, 2016

About us...

Welcome to Malaysia R User Group (myRUG)

+ Schedule a new Meetup

Upcoming | Past | Calendar

There are no upcoming Meetups

You can schedule one!

Schedule a Meetup

Recent Meetups

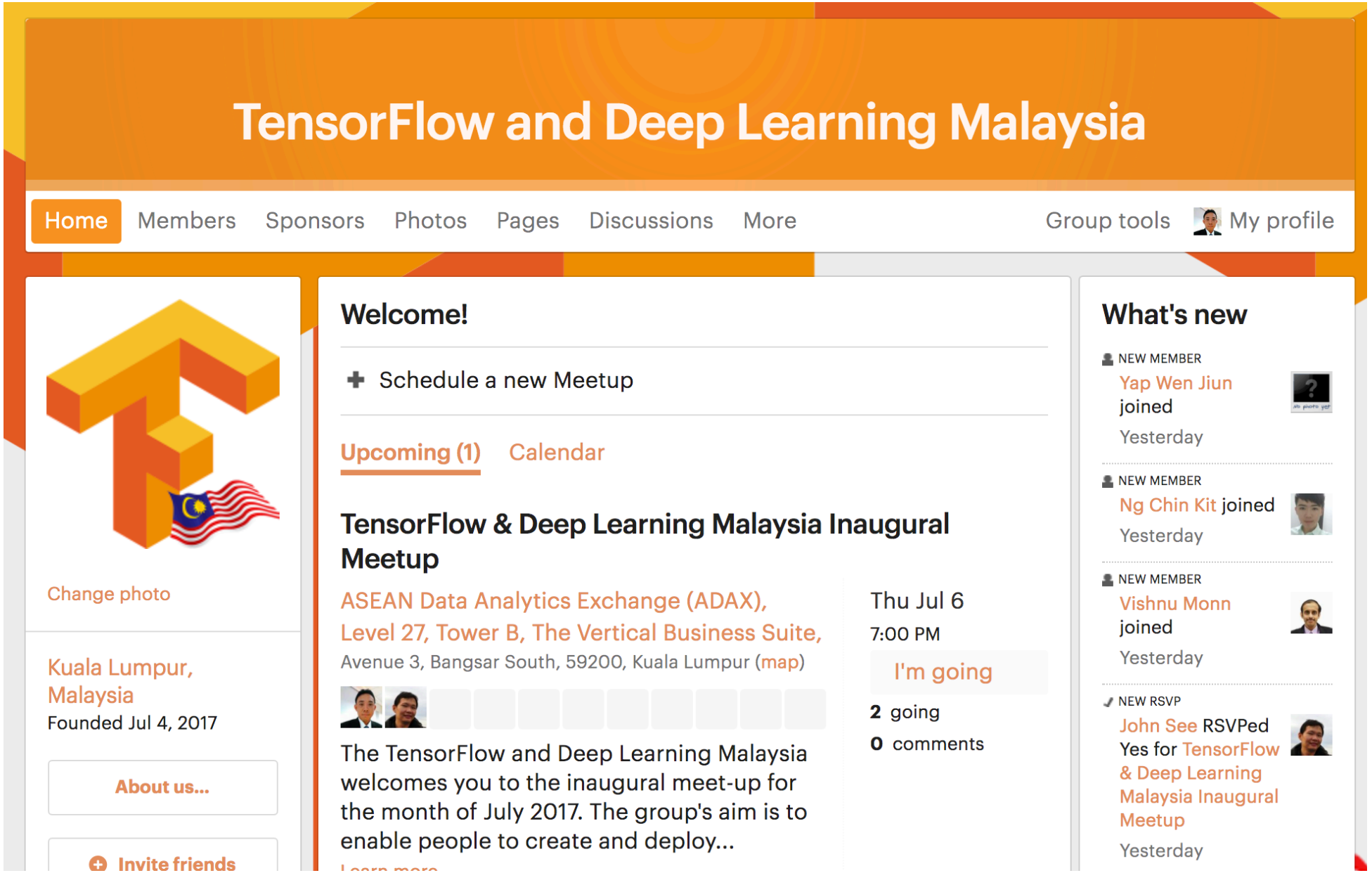
Oct 20, 2016 · 7:00 PM

Rate this Meetup

What's new

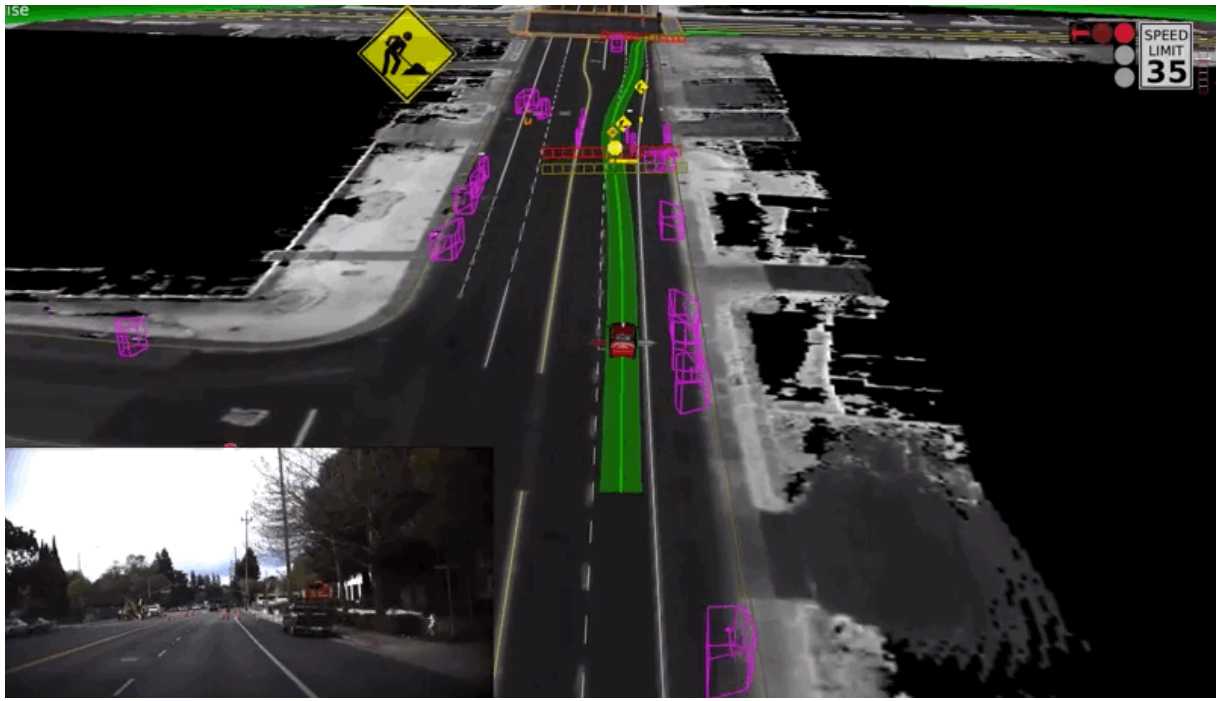
https://www.meetup.com/MY-RUserGroup/

https://www.facebook.com/groups/TensorFlowMY/



https://www.meetup.com/tensorflow-deep-learning-malaysia

Where are we now…



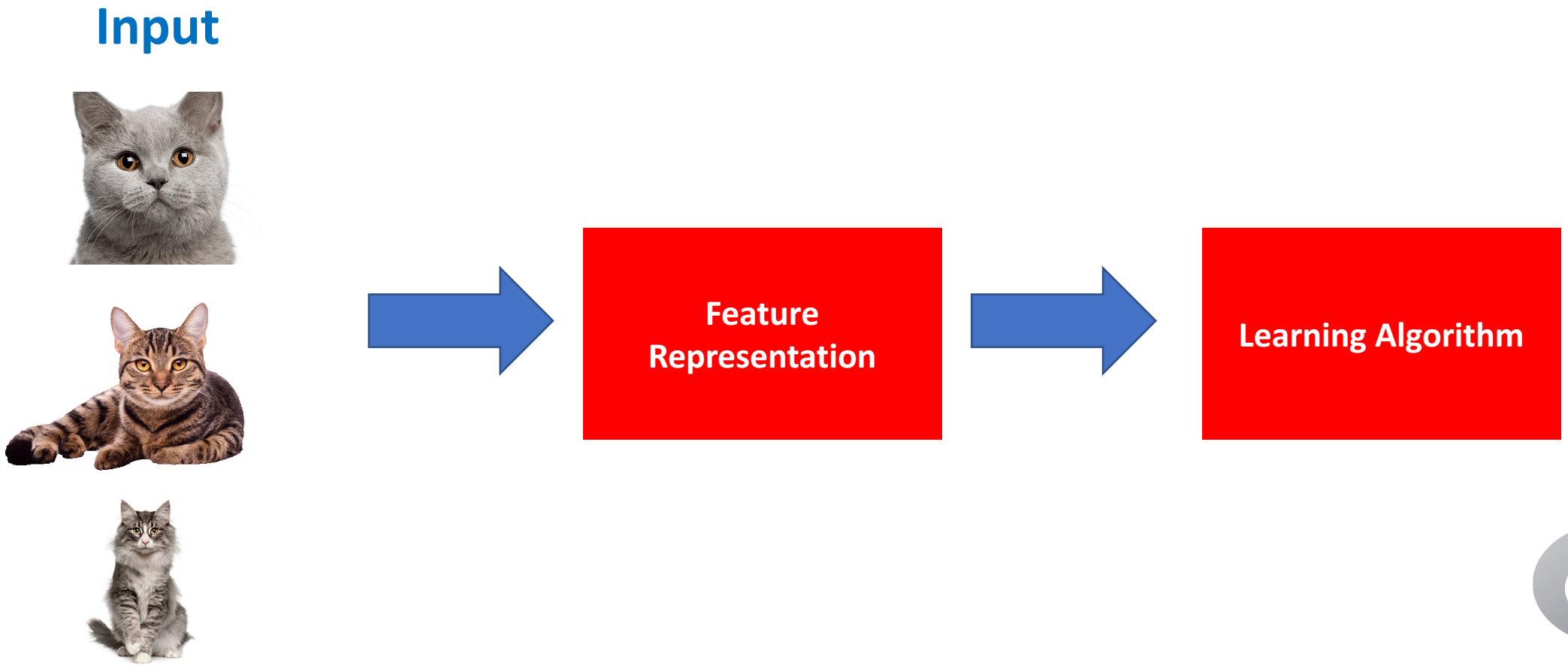
Where are we now..

Introduction

- In the past 10 years, machine learning and Artificial Intelligence (AI) have shown tremendous progress
- The recent success can be attributed to:
  - Explosion of data
  - Cheap computing cost - CPUs and GPUs
  - Improvement of machine learning models
- Much of the current excitement concerns a subfield of it called “Deep Learning”.

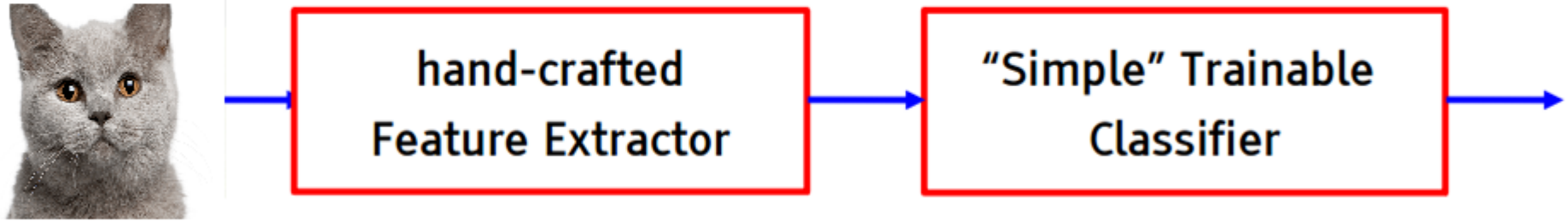
A brief history of Machine learning

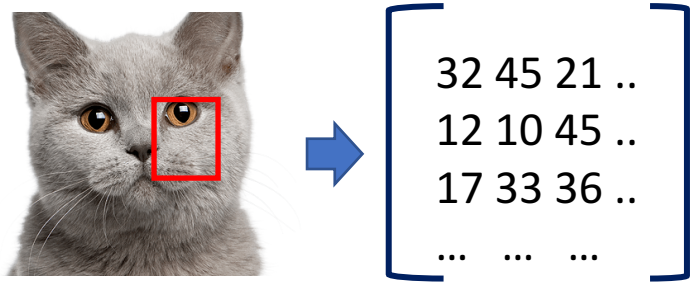
- Most of the machine learning methods are based on supervised learning

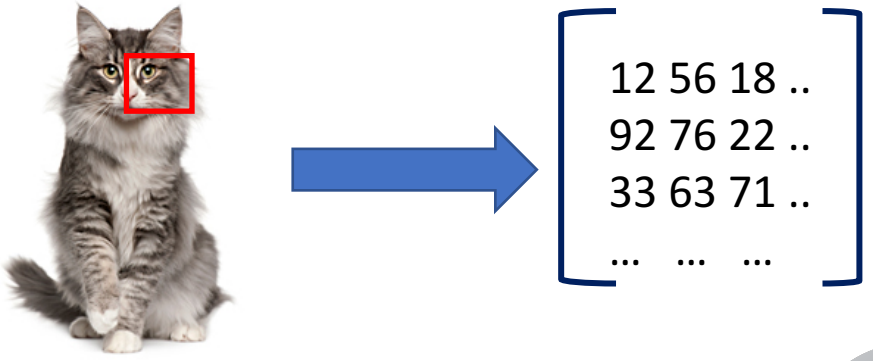


A brief history of Machine learning

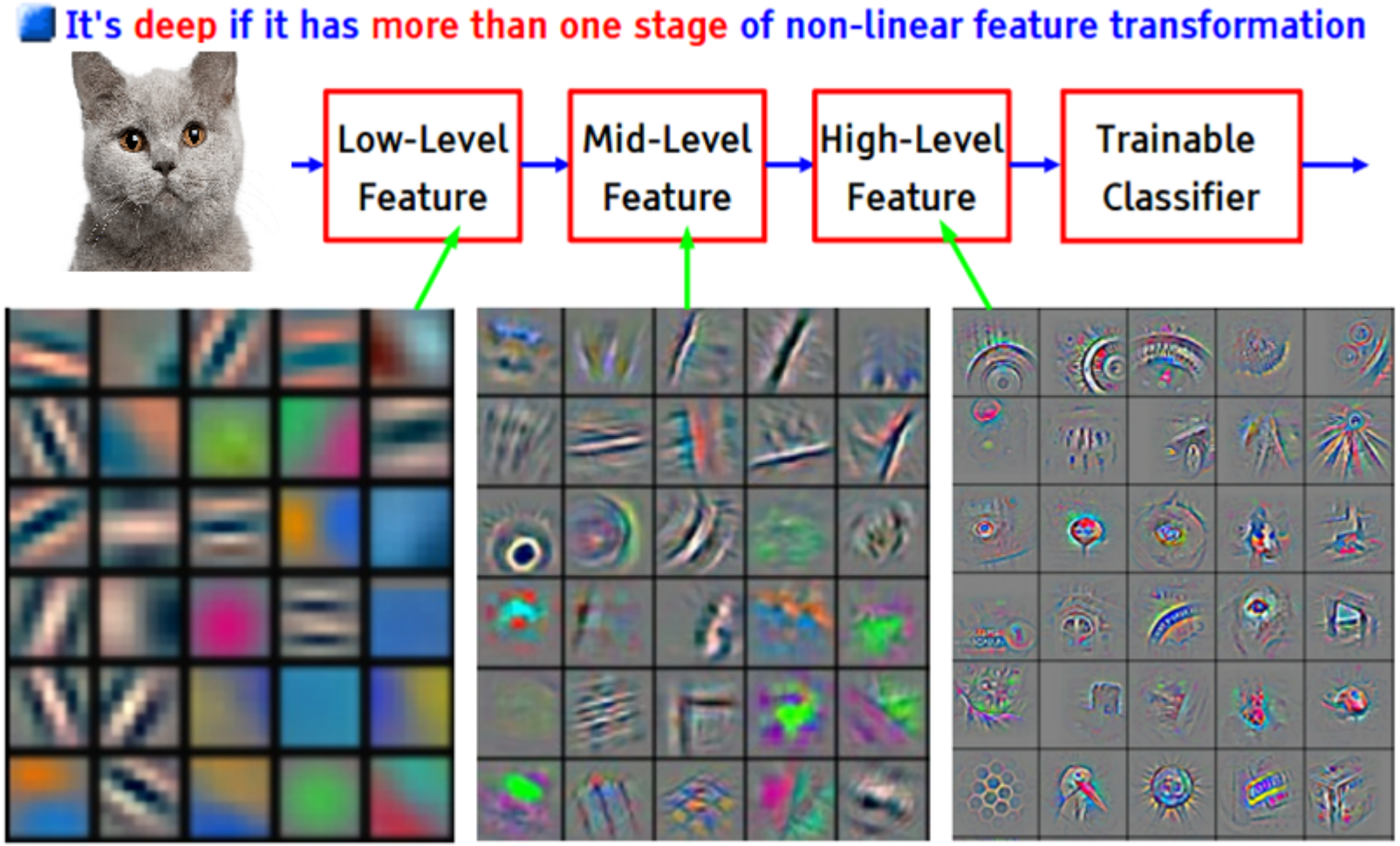
- **The traditional model of pattern recognition (since the late 50's)**
  - Fixed/engineered features (or fixed kernel) + trainable classifier



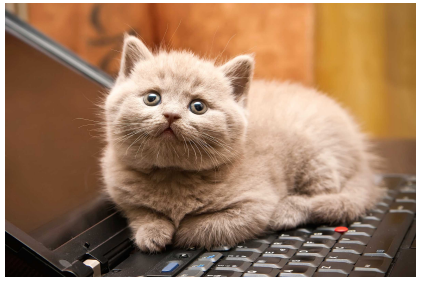


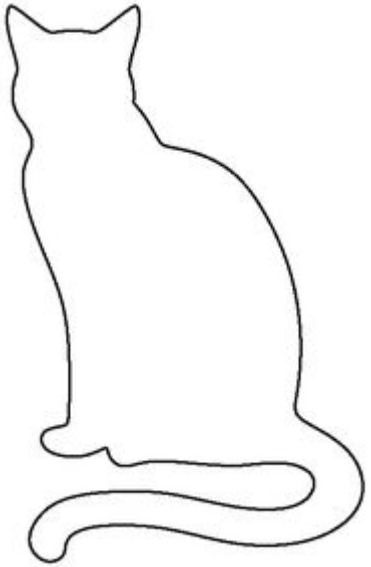


Features Training

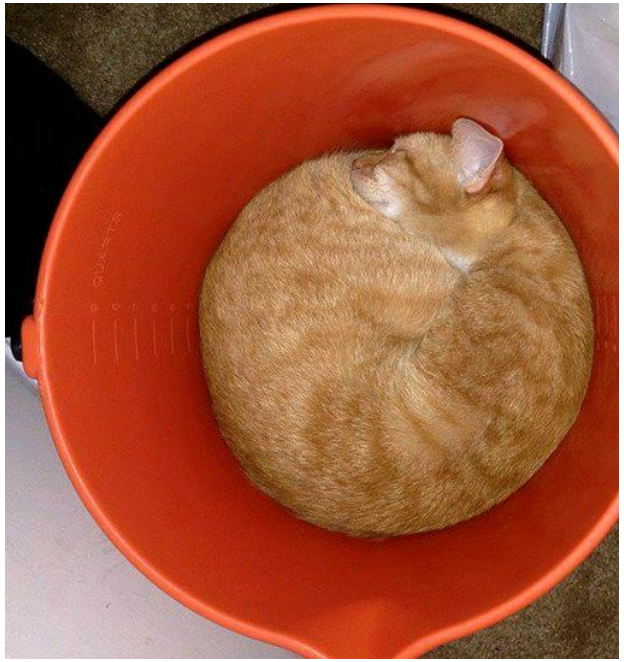


Eg. Features of a cat



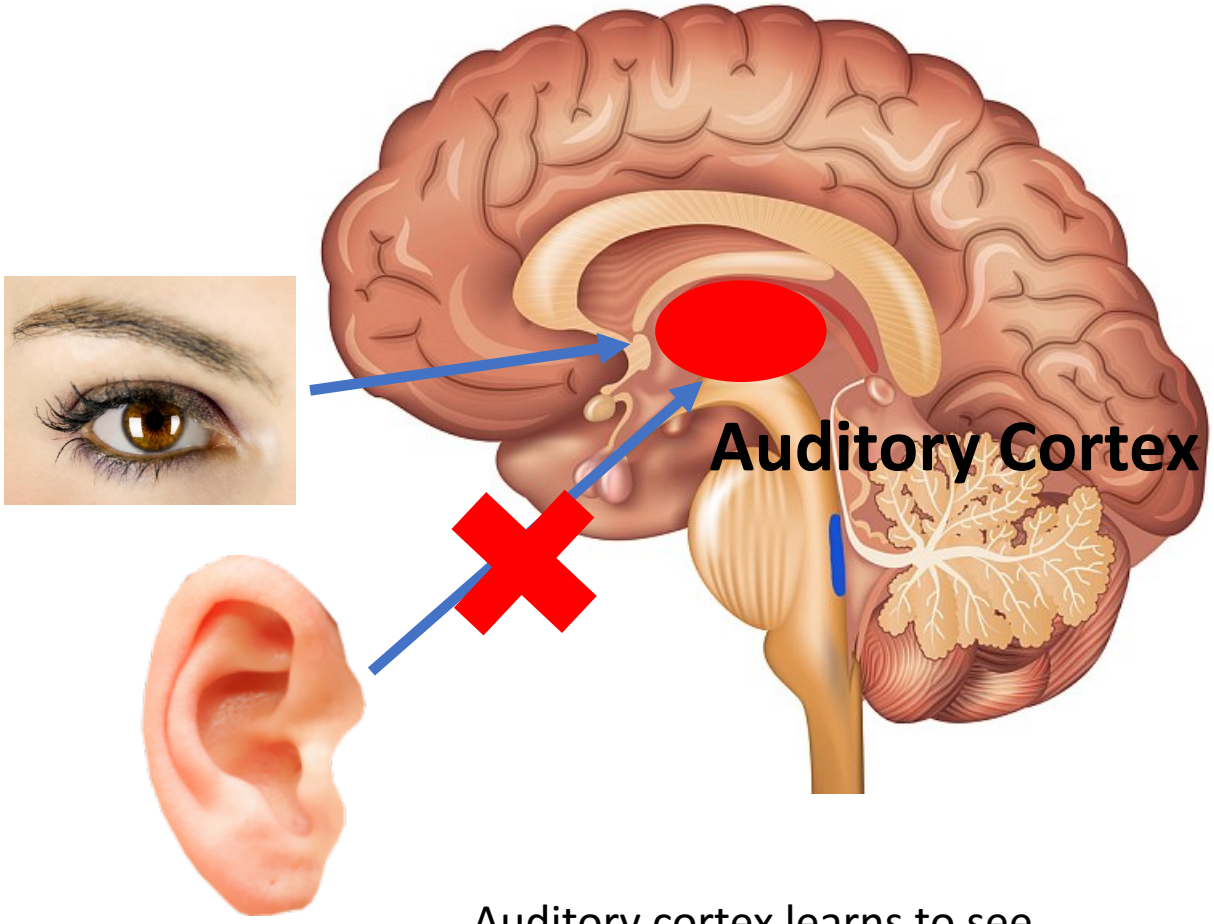


Cat?

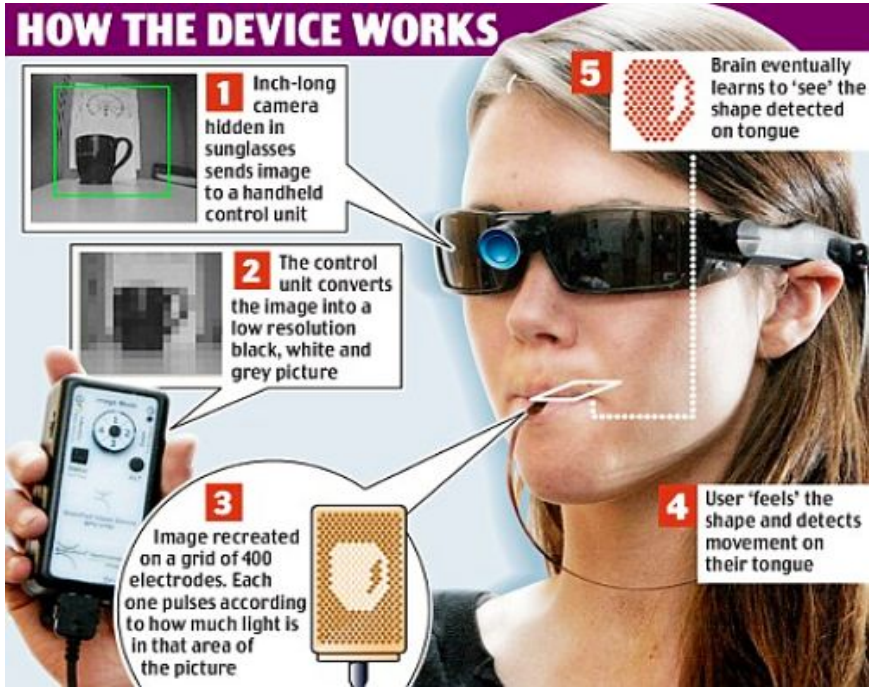


How can we do better?

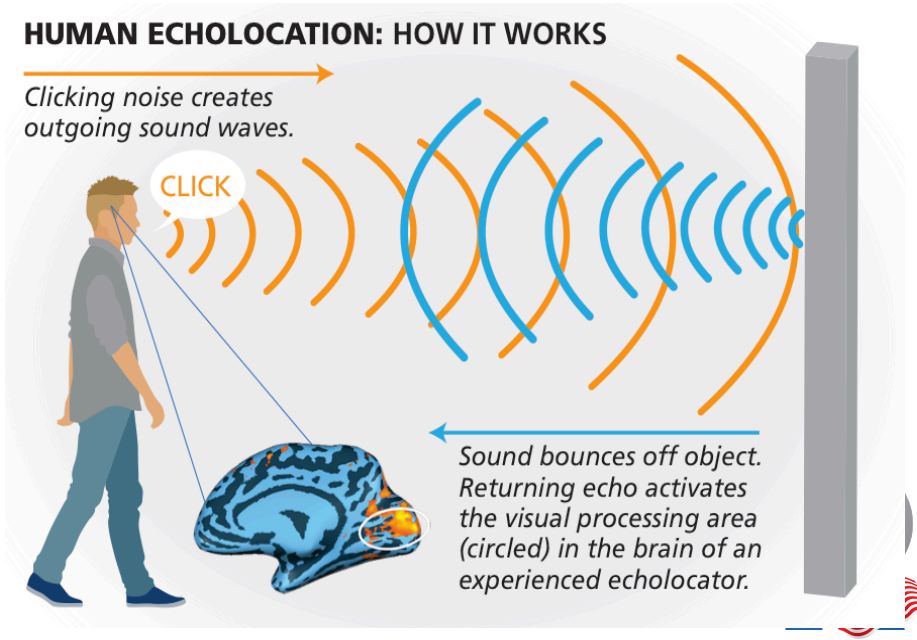
Human Brain



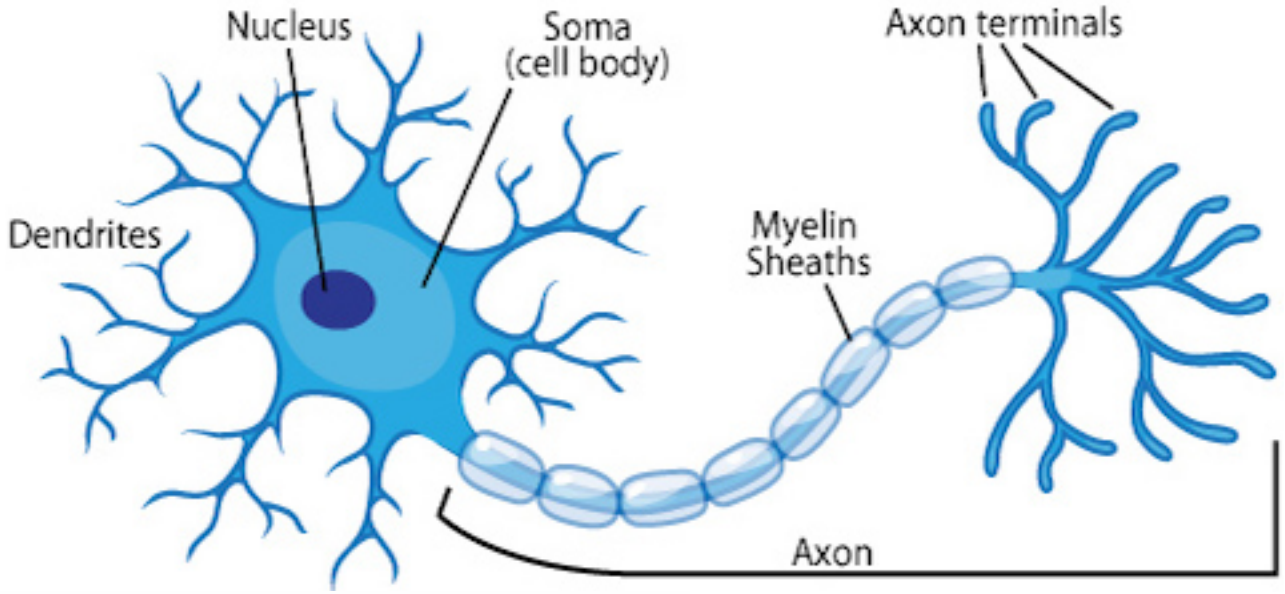
Auditory cortex learns to see. (Same rewiring process also works for touch/ somatosensory cortex.)

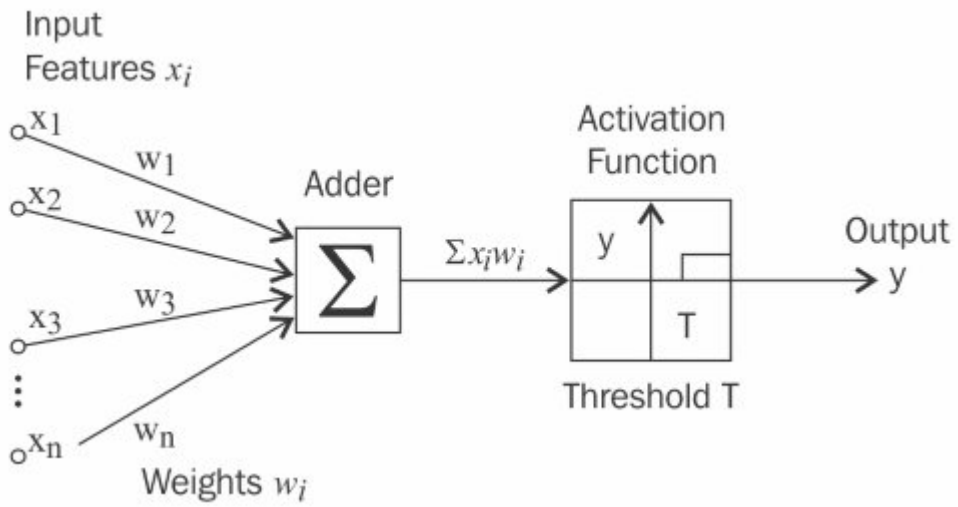


Seeing with tongue



Human Brain

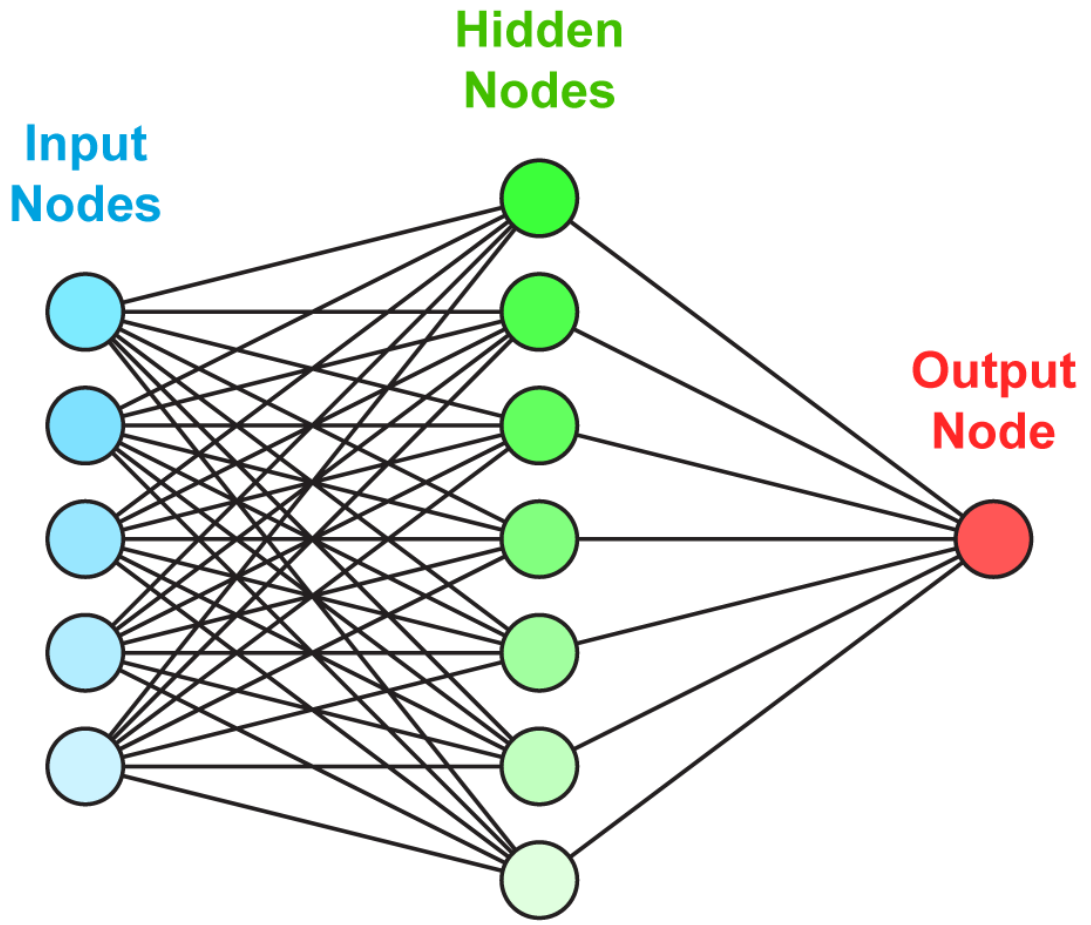




$$y = g\left(w_0 + \sum_{i=1}^{p} w_i x_i\right)$$

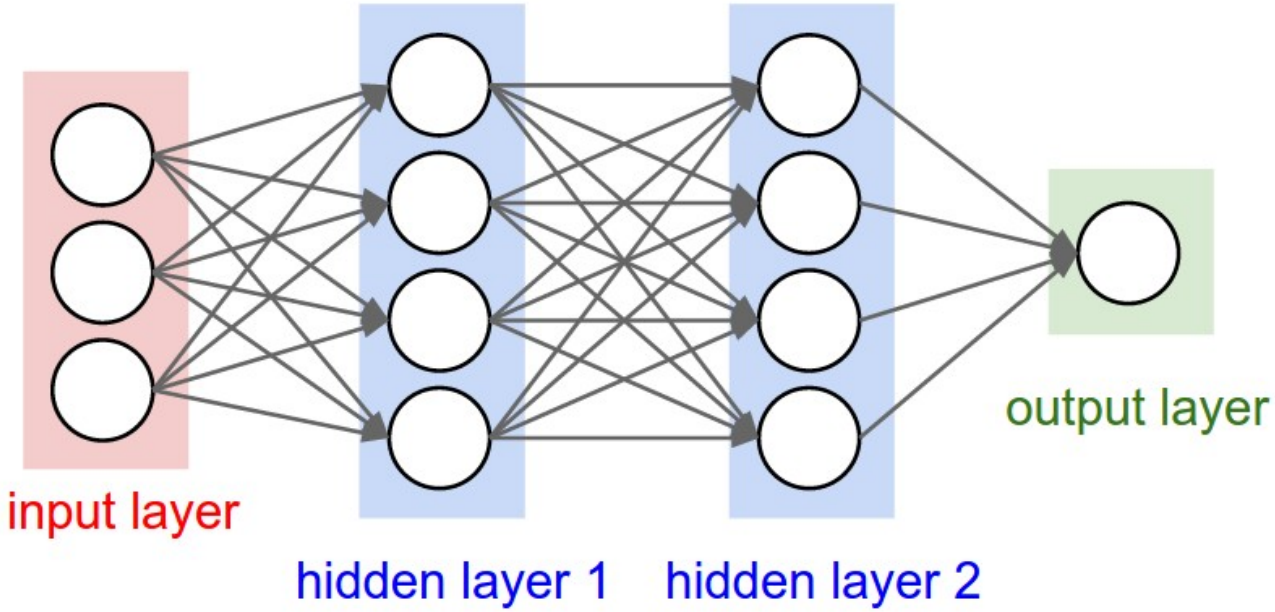
Neural Network

- Deep Learning is primarily about neural networks, where a network is an interconnected web of nodes and edges.
- Neural nets were designed to perform complex tasks, such as the task of placing objects into categories based on a few attributes.
- Neural nets are highly structured networks, and have three kinds of layers - an input, an output, and so called hidden layers, which refer to any layers between the input and the output layers.
- Each node (also called a neuron) in the hidden and output layers has a classifier.



Neural Network: Forward Propagation

- The input neurons first receive the data features of the object. After processing the data, they send their output to the first hidden layer.
- The hidden layer processes this output and sends the results to the next hidden layer.
- This continues until the data reaches the final output layer, where the output value determines the object's classification.
- This entire process is known as Forward Propagation, or Forward prop.

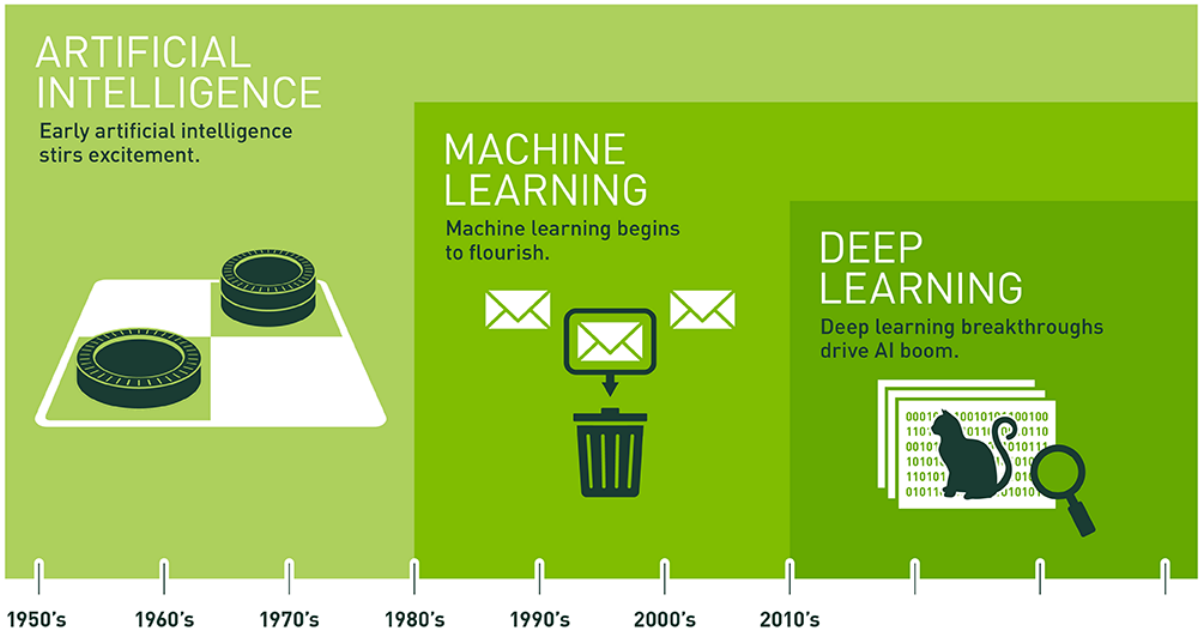


Neural Network: Backward Propagation

- To train a neural network over a large set of labelled data, you must continuously compute the difference between the network's predicted output and the actual output.
- This difference is called the cost, and the process for training a net is known as backpropagation, or backprop
- During backprop, weights and biases are tweaked slightly until the lowest possible cost is achieved.
- An important aspect of this process is the gradient, which is a measure of how much the cost changes with respect to a change in a weight or bias value.

Deep Learning

- Deep learning refers to artificial neural networks that are composed of many layers.
- It's a growing trend in Machine Learning due to some favorable results in applications where the target function is very complex and the datasets are large.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Deep Learning: Strengths

- **Robust**
  - No need to design the features ahead of time - features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- **Generalizable**
  - The same neural net approach can be used for many different applications and data types
- **Scalable**
  - Performance improves with more data, method is massively parallelizable

Deep Learning: Weaknesses

- Deep Learning **requires a large dataset**, hence long training period.
- In term of cost, Machine Learning methods like SVMs and other tree ensembles are very easily deployed even by relative machine learning novices and can usually get you reasonably good results.
- Deep learning methods **tend to learn everything**. It's better to encode prior knowledge about structure of images (or audio or text).
- The learned features are often **difficult to understand**. Many vision features are also not really human-understandable (e.g, concatenations/combinations of different features).
- Requires **a good understanding of how to model** multiple modalities with traditional tools.

Deep Learning: Applications

Deep Learning Libraries

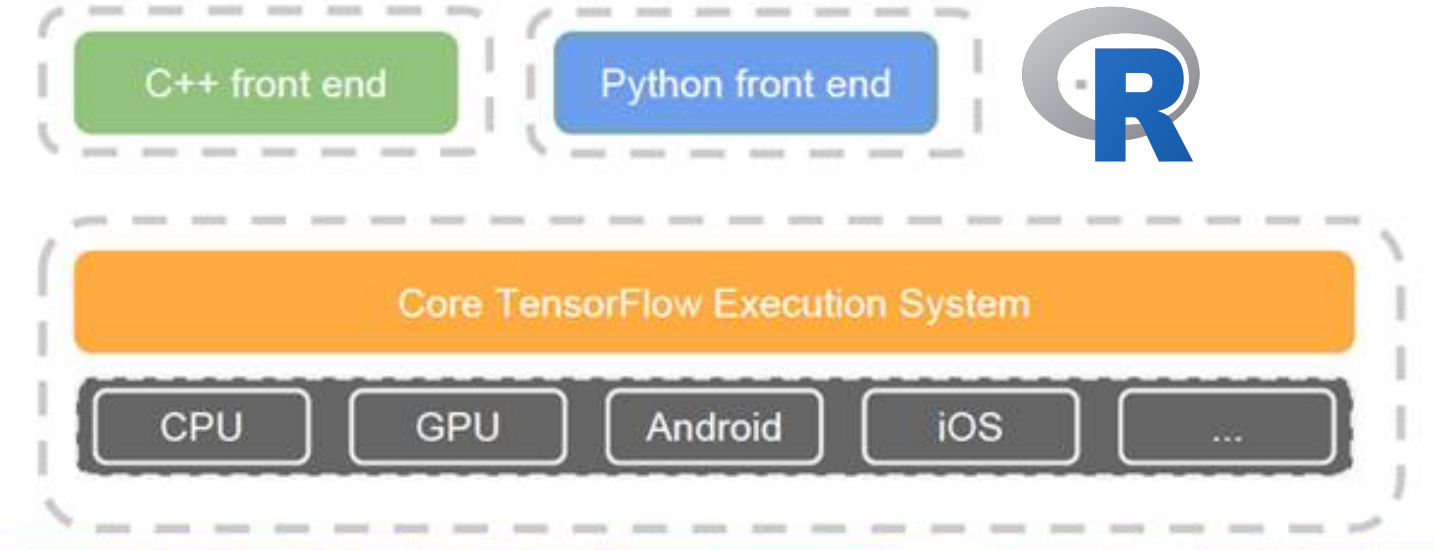
- MXNet: The R interface to the MXNet deep learning library.
- darch: An R package for deep architectures and restricted Boltzmann machines.
- deepnet: An R package implementing feed-forward neural networks, restricted Boltzmann machines, deep belief networks, and stacked autoencoders.
- Tensorflow for R: The tensorflow package provides access to the complete TensorFlow API from within R.
- h2o: The R interface to the H2O deep-learning framework.
- Deepwater: GPU Enabled Deep Learning using h2o platform

TensorFlow

- URL: https://www.tensorflow.org/
- Released under the open source license on November 9, 2015
- Current version 1.2
- Open source software library for numerical computation using data flow graphs
- Originally developed by Google Brain Team to conduct machine learning and deep neural networks research
- General enough to be applicable in a wide variety of other domains as well
- TensorFlow provides an extensive suite of functions and classes that allow users to build various models from scratch.

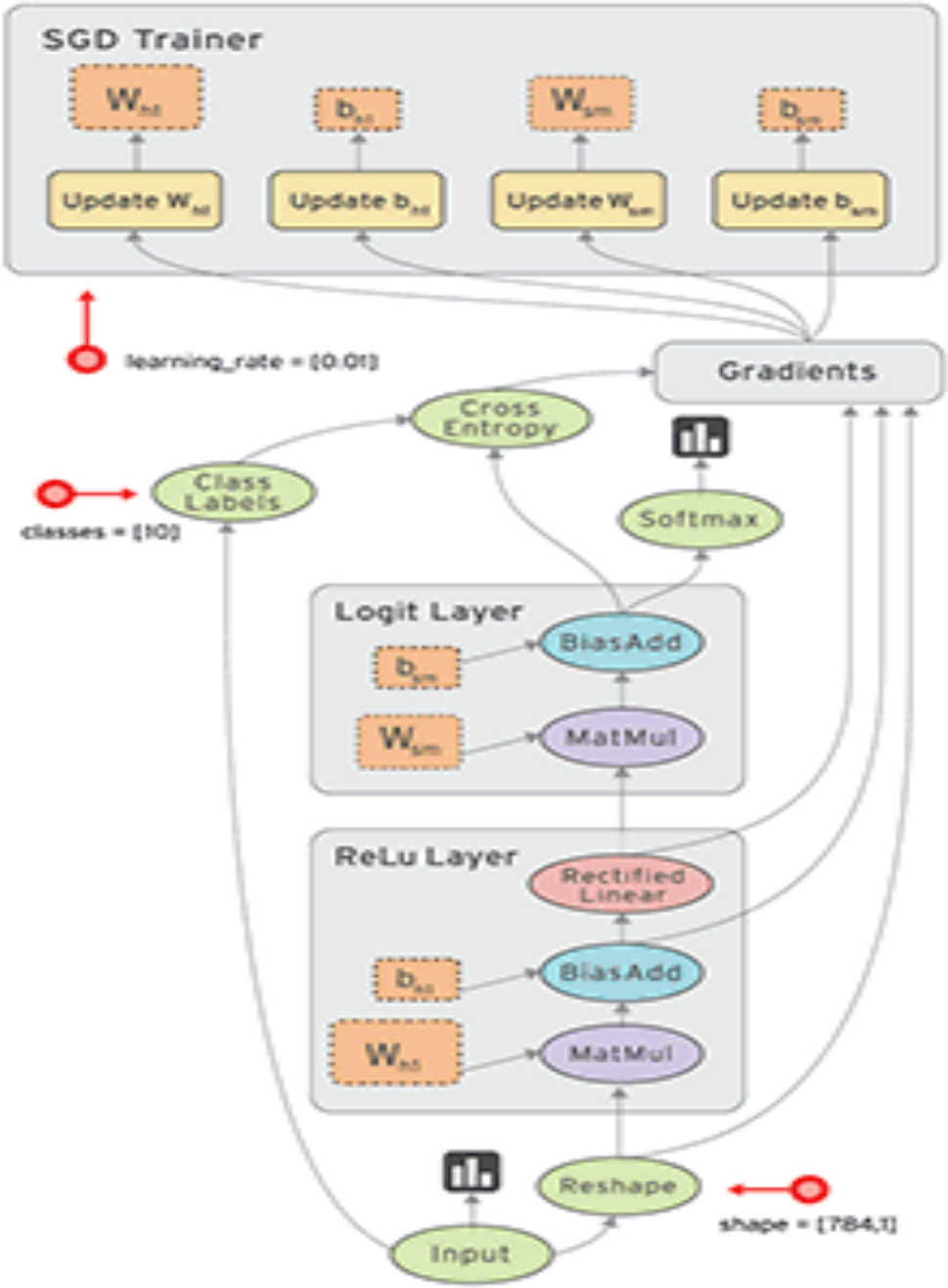
TensorFlow architecture

- Core in C++
  - Low overhead
- Different front ends for specifying/driving the computation
  - Python, C++, R and many more



TensorFlow Graph

- TensorFlow is based on computation data flow graph
- TensorFlow separates definition of computations from their execution



TensorFlow + R

- The TensorFlow API is composed of a set of Python modules that enable constructing and executing TensorFlow graphs.
- The tensorflow package provides access to the complete TensorFlow API from within R.
- URL: https://tensorflow.rstudio.com/

```
devtools::install_github("rstudio/tensorflow")
library(tensorflow)
install_tensorflow()
```

MXNET

- **Founded by:** Uni. Washington & Carnegie Mellon Uni (~ 2 years old)
- **Supports most OS**: Runs on Amazon Linux, Ubuntu/Debian, OS X, and Windows OS
- **State of the art model support:** Flexible and efficient GPU computing and state-of-art deep learning i.e. CNN, LSTM to R.
- **Ultra Scalable:** Seamless tensor/matrix computation with multiple GPUs in R.
- **Ease of Use:** Construct and customize the state-of-art deep learning models in R, and apply them to tasks, such as image classification and data science challenges
- **Multi-language:** Supports the Python, R, Julia and Scala languages
- **Ecosystem**: Vibrant community from Academia and Industry
- URL: http://mxnet.io/

Amazon + MXNET

siliconANGLE NEWS PEOPLE RESEARCH Wikibon TheCUBE SIGN IN

Amazon dives deeper into deep learning with MXNet framework

BY ERIC DAVID

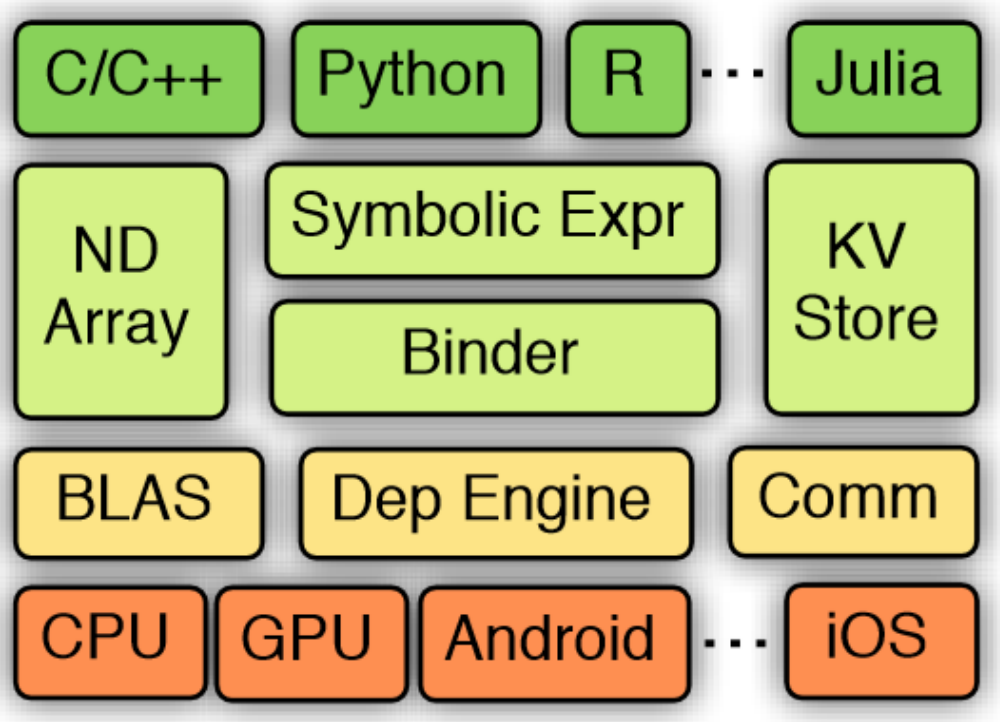
UPDATED 14:45 EST . 22 NOVEMBER 2016

Nearly every major tech company, from Google Inc. to Facebook Inc., has been making deep learning an increasingly important aspect of their businesses, and today Amazon.com Inc. got into the game.

The company, whose Amazon Web Services is the leader in cloud computing, announced that it has chosen the MXNet software framework on which it will build its own foundation for deep learning, a branch of artificial intelligence that attempts to emulate in software the way the brain learns.

MXNET Architecture

- You can specify the context of the function to be executed within. This usually includes whether the function should be run on a CPU or a GPU, and if you specify a GPU, which GPU to use.



MXNET R Package

- The R Package can be downloaded using the following commands:

```
install.packages("drat",
repos="https://cran.rstudio.com")
drat:::addRepo("dmlc")
install.packages("mxnet")
```

MXNET + R

```
#configure another network
data <- mx.symbol.Variable("data")
fc1 <- mx.symbol.FullyConnected(data, name = "fc1", num_hidden=10) #1st hidden layer
act1 <- mx.symbol.Activation(fc1, name = "sig", act_type="relu")
fc2 <- mx.symbol.FullyConnected(act1, name = "fc2", num_hidden=2) #2nd hidden layer
out <- mx.symbol.SoftmaxOutput(fc2, name = "soft")
```

Thanks!

Questions?

@kuanhoong

https://www.linkedin.com/in/kuanhoong

kuanhoong@gmail.com

DEMO

Slides and Codes are available:
http://github.com/kuanhoong/deeplearning-r