

# Deep Learning on Azure

**Tim Scarfe**



Data Solution Architect / Data Scientist



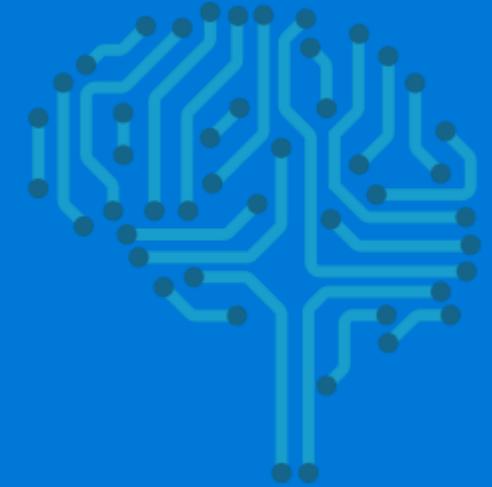
[tim.scarfe@microsoft.com](mailto:tim.scarfe@microsoft.com) / [@ecsquendor](https://twitter.com/ecsquendor)

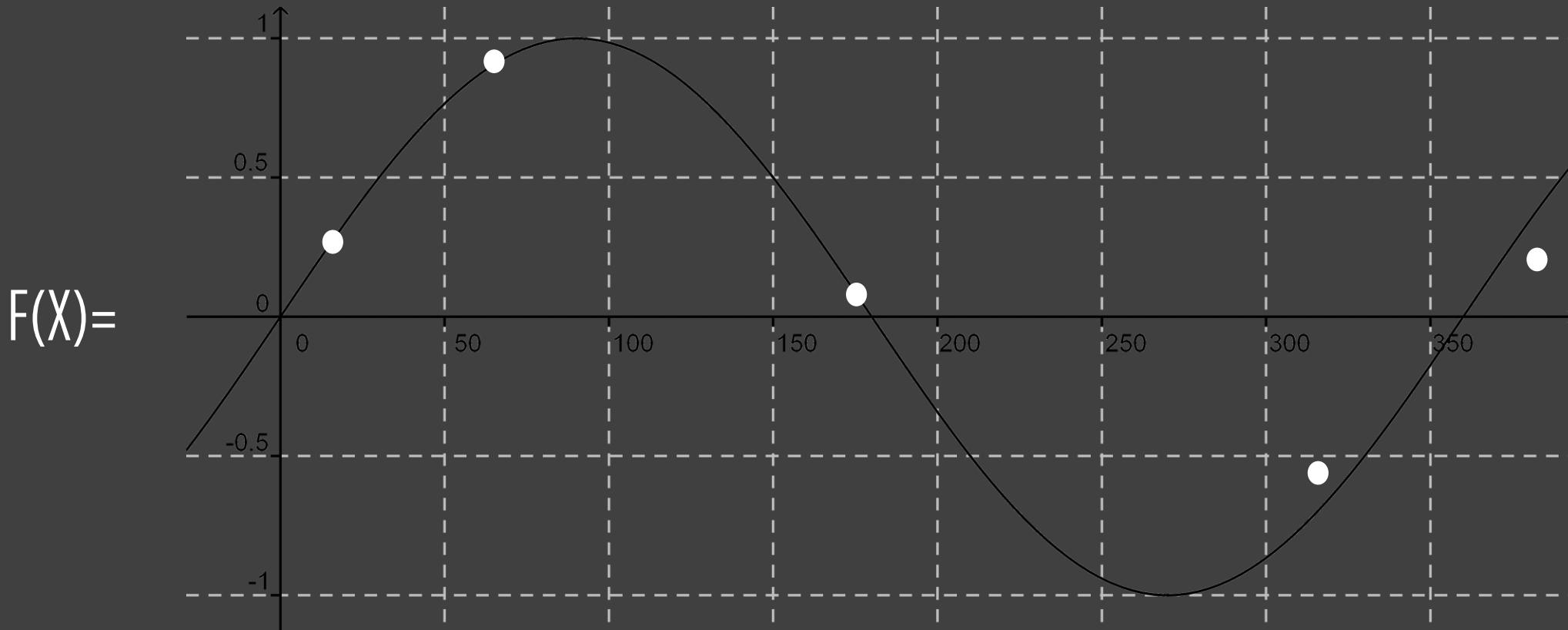




# WHAT IS MACHINE LEARNING?

Machine learning is about a machine learning from previous experience so that it can perform some task better *based on that experience*.



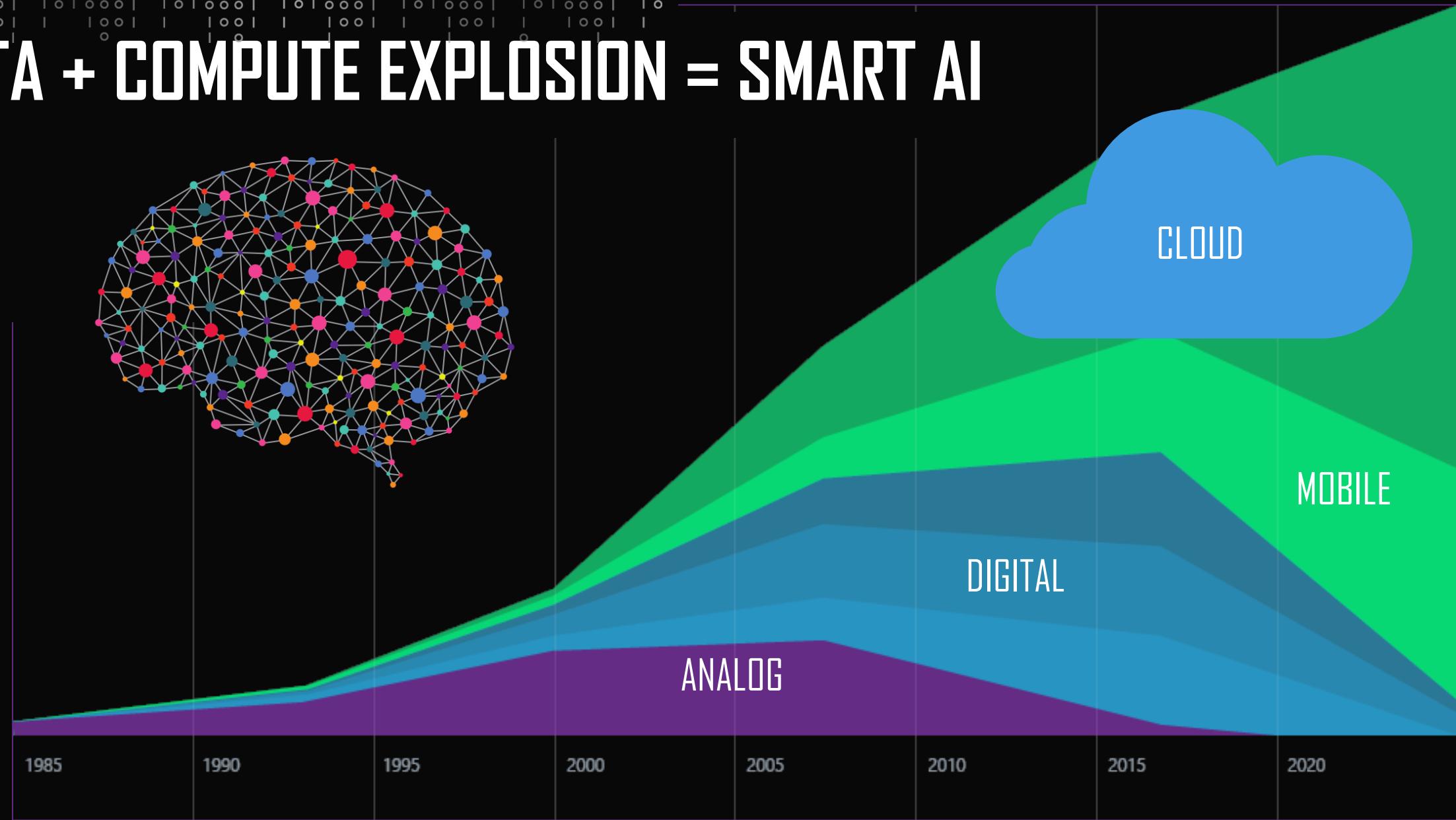


Machine learning approximates a function, which we may only have a few examples of.



# A SIMPLE SINEFUNCTION

# DATA + COMPUTE EXPLOSION = SMART AI



NICK BOSTROM

# SUPERINTELLIGENCE

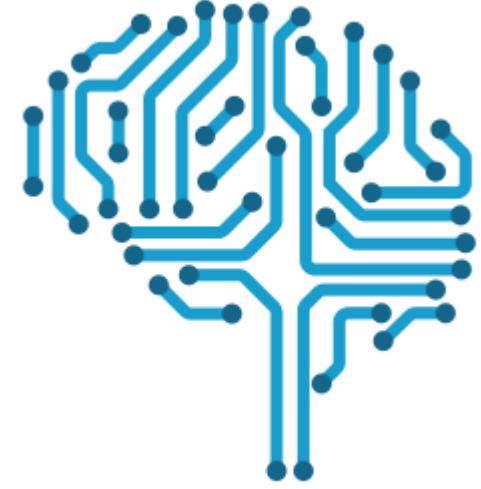
Paths, Dangers, Strategies



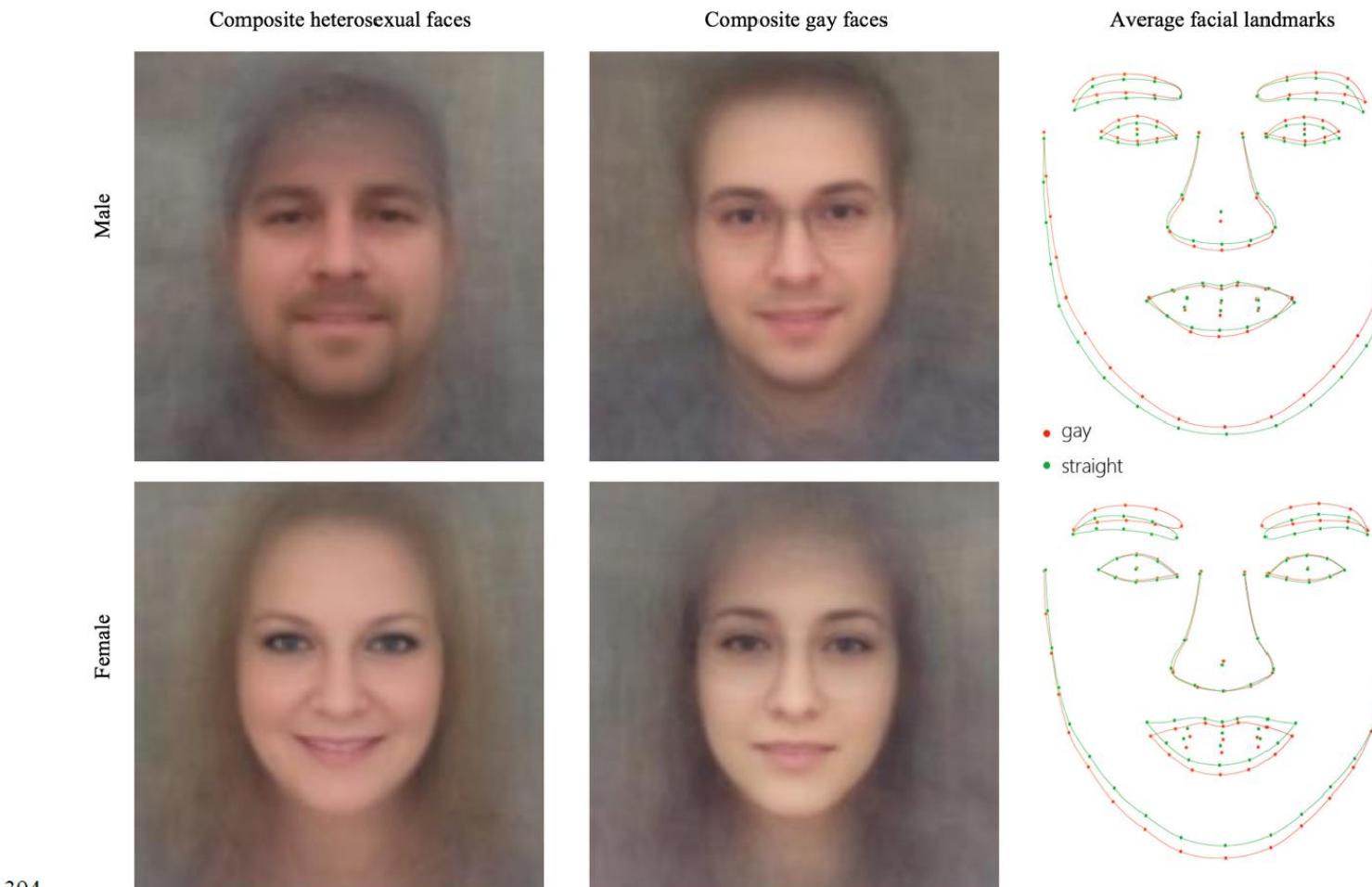
DON'T WORRY ABOUT GENERAL INTELLIGENCE



1. Discrimination
2. Opaque AI
3. Data is not neutral
4. Manipulating markets and consumers/voters
5. Lack of human connection
6. Engineers are not philosophers (moral reasoning)
7. Privacy
8. Innocent till proven guilty?



DO WORRY (A LOT) ABOUT ETHICS IN AI



394

395 *Figure 4.* Composite faces and the average facial landmarks built by averaging faces classified as most and least likely to be gay.

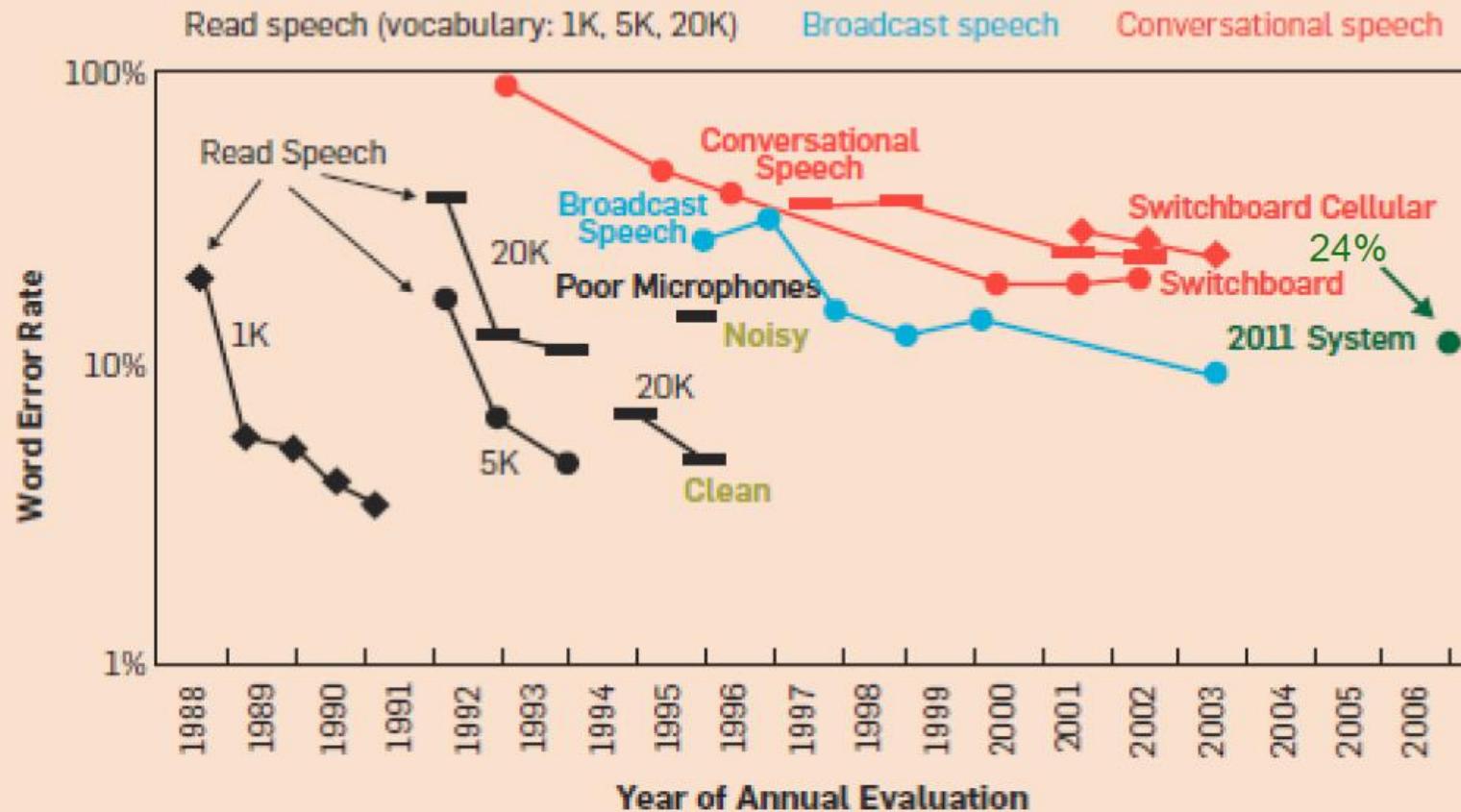
2017 Wang, Y., & Kosinski, M.

# PHYSIOGNOMY IS BACK?



**"Our goal is to democratise AI to empower every person and every organisation to achieve more."**

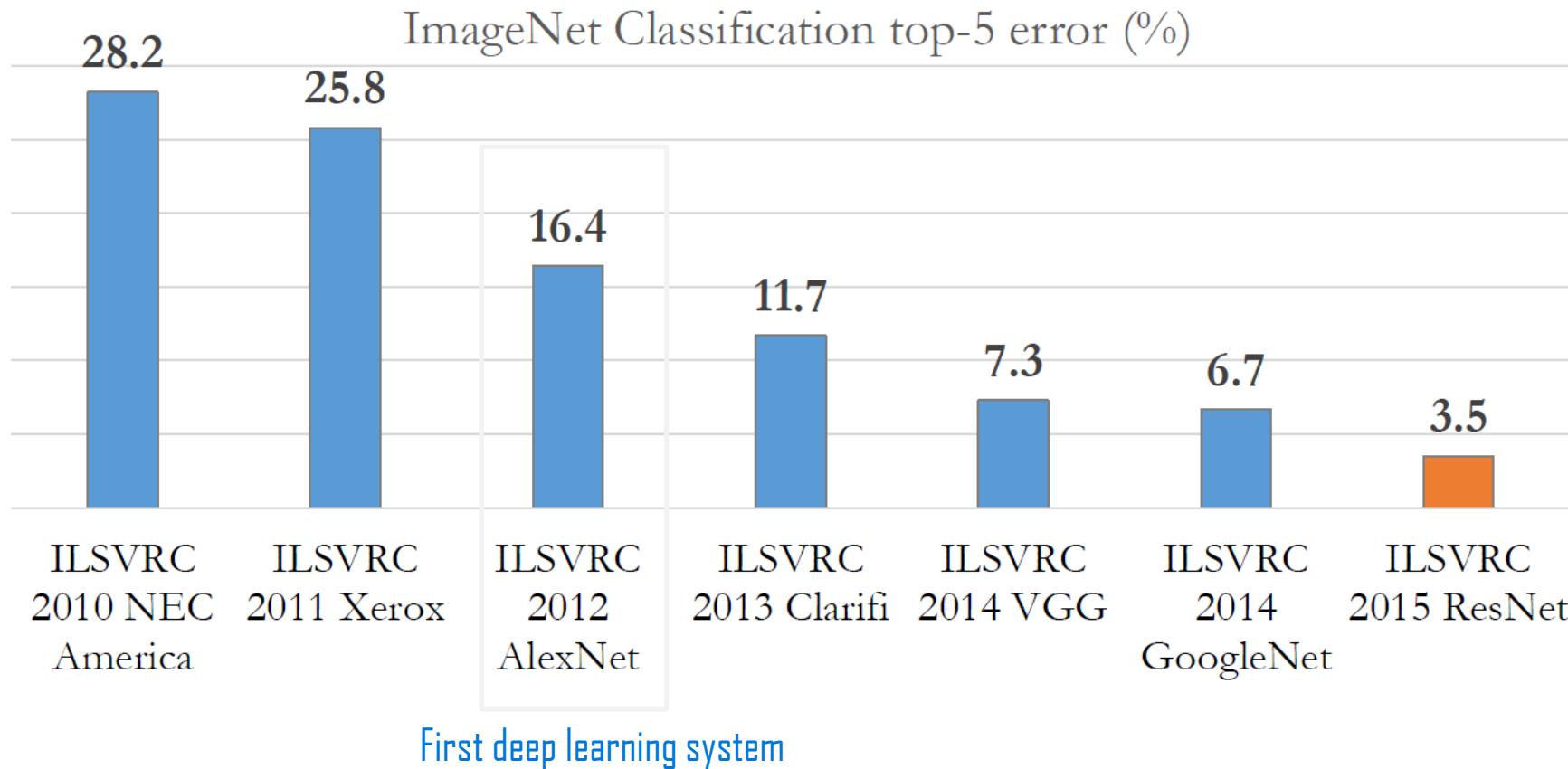
Satya Nadella



2017: ~5%!  
Human error: ~5%

# IMPROVEMENTS IN SPEECH RECOGNITION

# IMPROVEMENTS IN COMPUTER VISION



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



# Machine Learning/AI Stack at Microsoft



Cognitive Services



AzureML



R Server/MMLSpark



CNTK

Basic discriminative models

Novel prediction architectures

SaaS (REST)

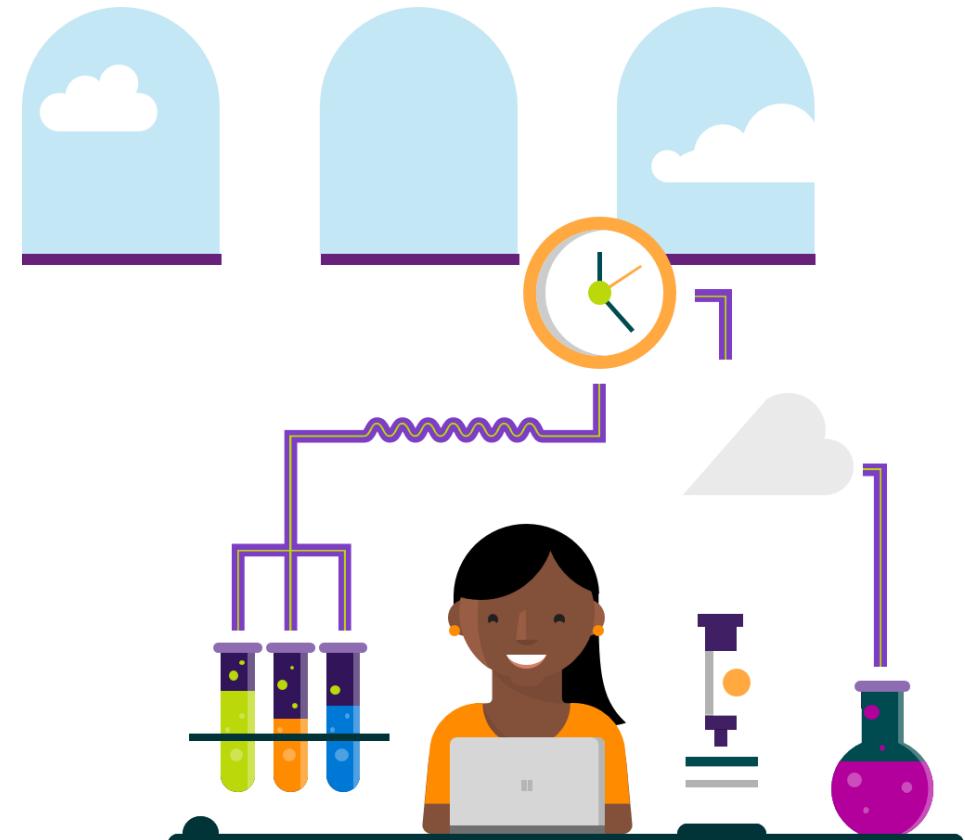
PaaS (Drag/Drop)

Code-first, hybrid  
execution model



# PROJECT BRAINWAVE

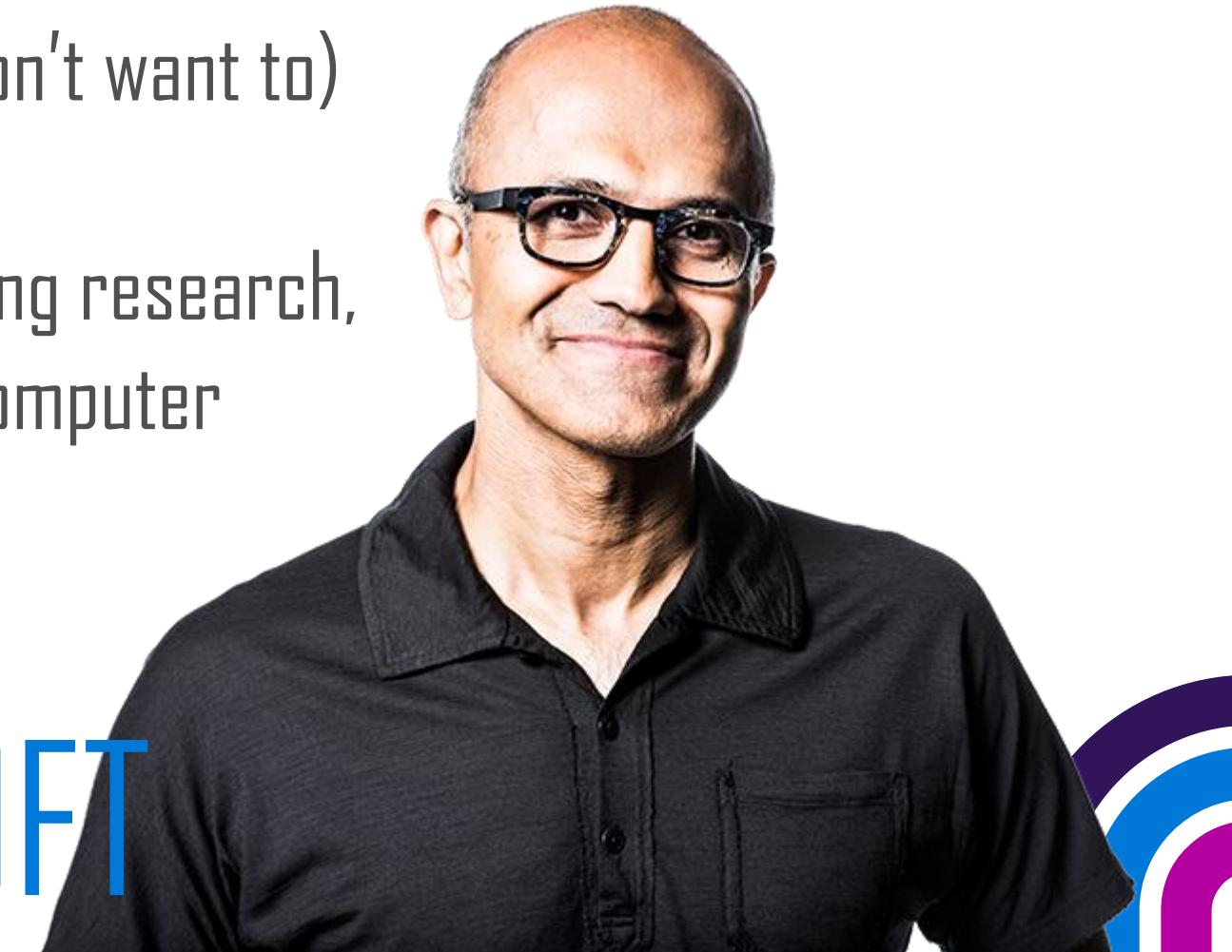
<https://batchaitraining.azure.com>



# AZURE BATCH AI TRAINING SERVICE

- We are #1 contributors to open source
- Platinum member of the Linux foundation
- We support all main deep learning frameworks  
(you don't have to use CNTK if you don't want to)
- CNTK is 100% open source
- We are world-leading on deep learning research,  
especially speech recognition and computer  
vision.

# THE NEW MICROSOFT



# Images/Sequences

## Supervised

Tell the machine the correct answer

Classification



## TYPES OF MACHINE LEARNING

Regression

Anomaly detection

Embeddings

Agent-based learning

## Unsupervised

Tell the machine nothing, let it observe the world

Clustering

Dimensionality Reduction

## Reinforcement Learning

Only tell machine if it was right or wrong

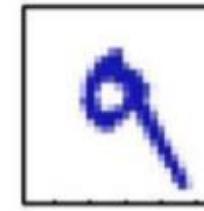
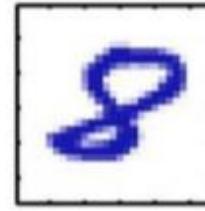
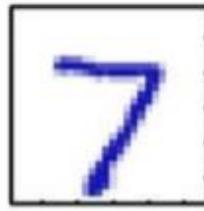
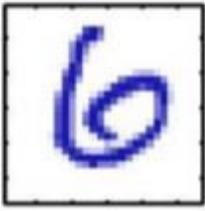
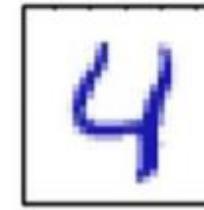
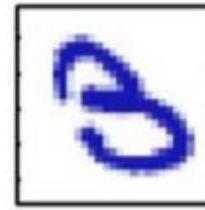
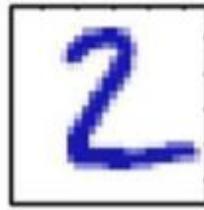
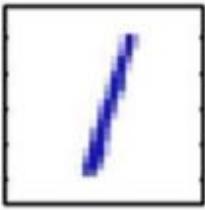
Autonomous cars

- Approximate a function which maps from signals (image) to labels (has-cat)
- This “decision function” can predict missing labels (has-cat) on new, previously unseen signals.



# MANDATORY CAT EXAMPLE

# MNIST Digit Classification



Images are 28 x 28 pixels

Represent input image as a vector  $\mathbf{x} \in \mathbb{R}^{784}$

Learn a classifier  $f(\mathbf{x})$  such that,

$$f : \mathbf{x} \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Signals

Labels

# DISTILLED CONCEPTS OF DEEP LEARNING



- The networks have **many levels** of depth
- Machine learns a hierarchy of representations
- **No feature extraction required**



Traditional ML

Hand crafted features

Feature Extractor



Trainable Classifier



Deep Learning

Representations are hierarchical and trained automatically

Low Level Features



Mid Level Features



High Level Features

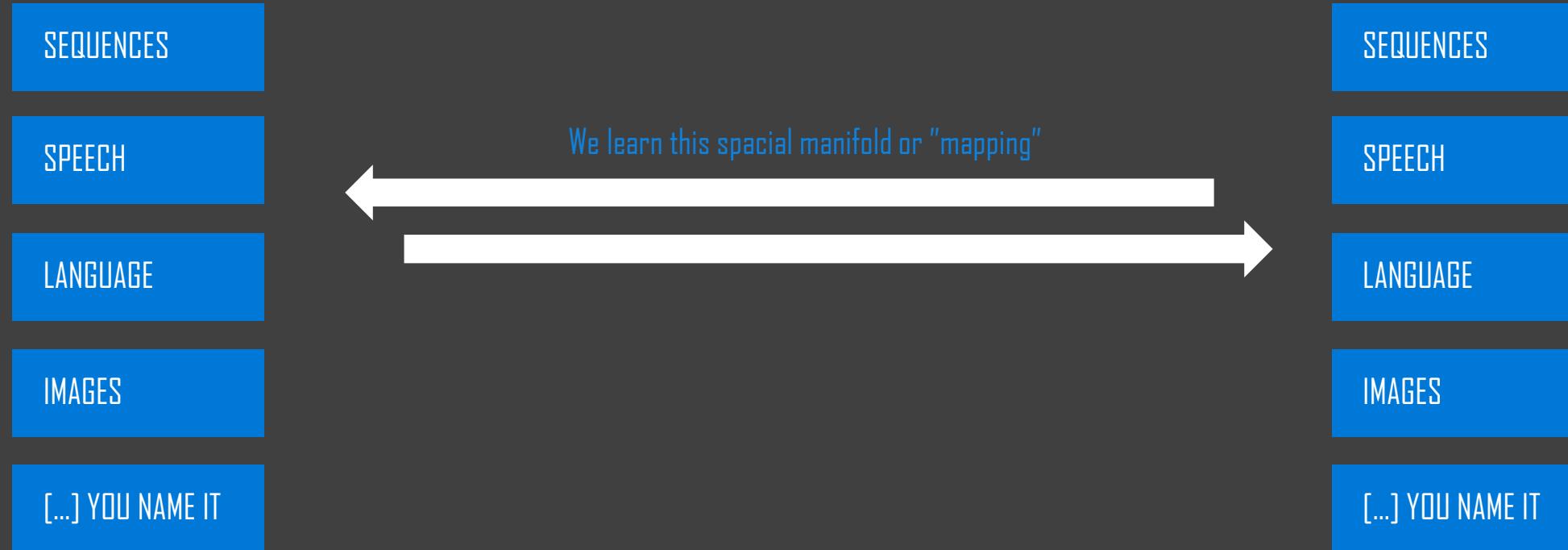


Trainable Classifier



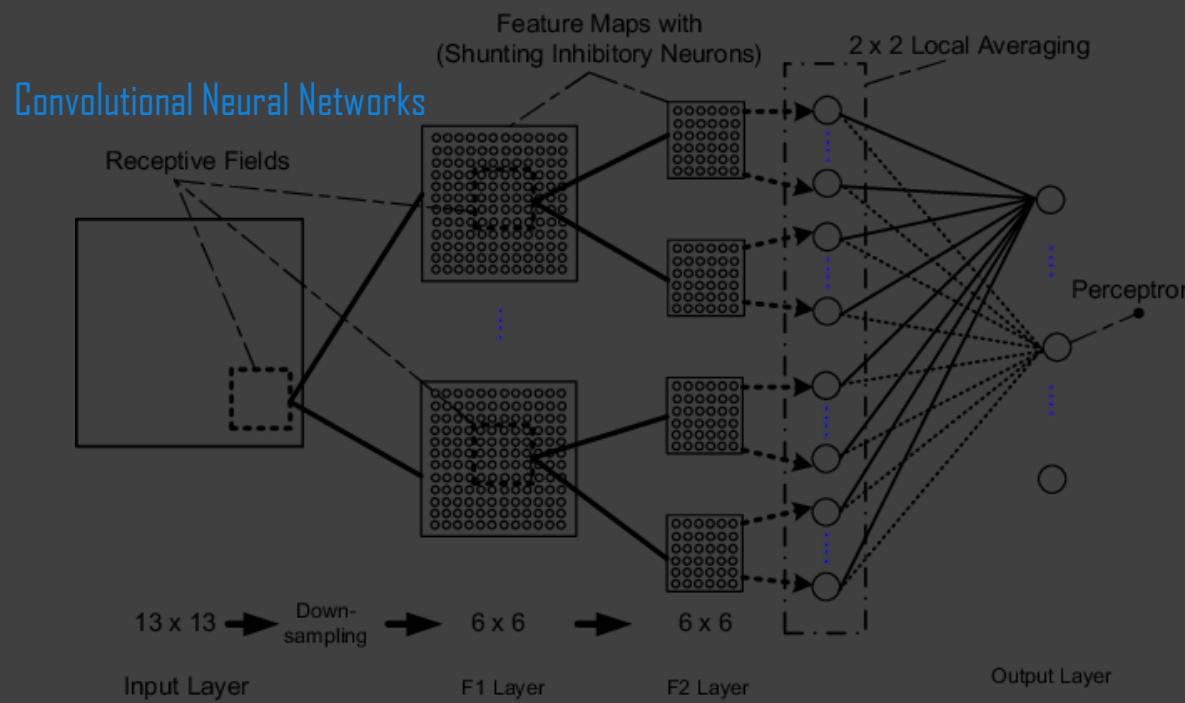
**"DEEP"** = THE ENTIRE MACHINE IS TRAINABLE

- Unlike other (frequentist) ML algorithms; you can map from *anything* to *anything*

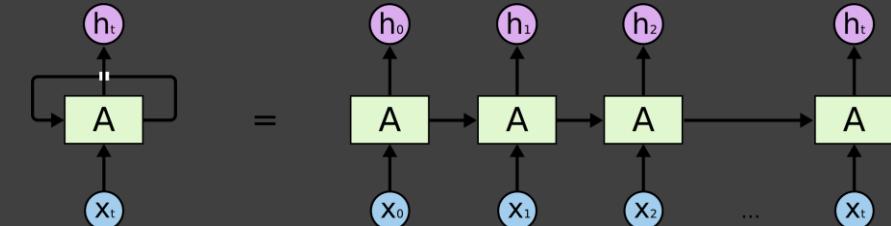


# UNIVERSAL FUNCTION APPROXIMATORS

- Unlike other algorithms, NNs can encode useful and obvious relationships in the data domain
  - Local spatial dependencies (vision)
  - Time dependencies (language, speech)

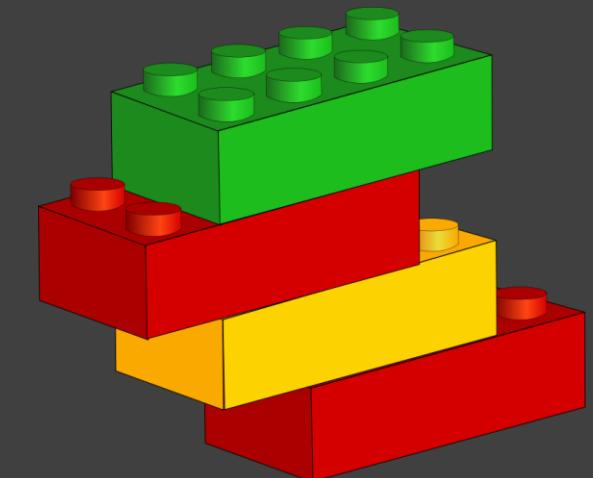


### Recurrent Neural Networks



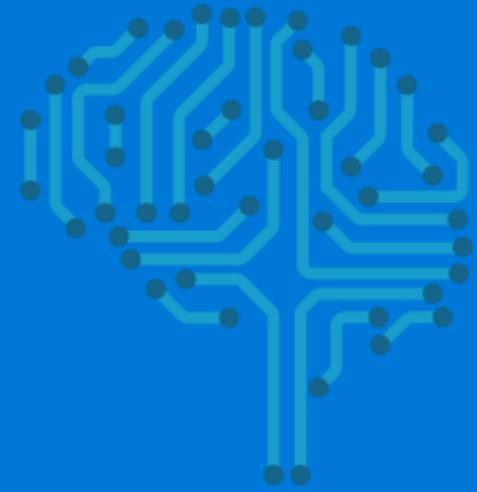
# CAPTURE DATA-DOMAIN FEATURES DIRECTLY

- Pre-existing deep learning components can be composed together in an entirely new way, like existing software development
- Deep Learning research is very applied compared to other ML research
- Most innovations are based on finding new architectures or composing existing networks together in new ways
- The people who work on DL tend to be software type people
- Frameworks make DL ubiquitously accessible to anyone who knows a bit of Python, and an end-to-end prediction architecture is contained inside one modality
- Very innovative architectures are possible with deep learning, for example GANs, Autoencoders (explained later).



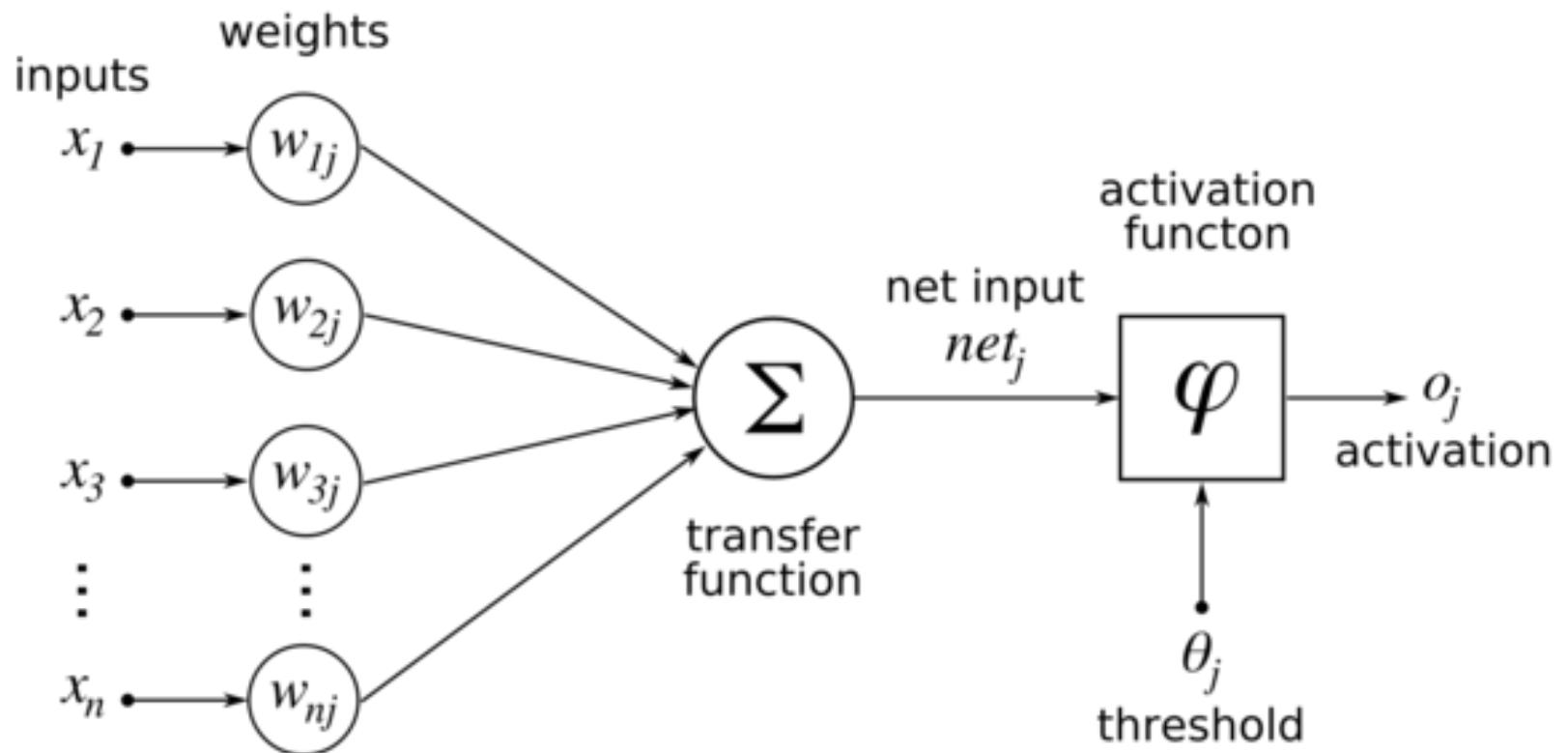
# COMPOSABILITY / MORE LIKE SOFTWARE THAN SCIENCE

# DEEP LEARNING CONCEPTS

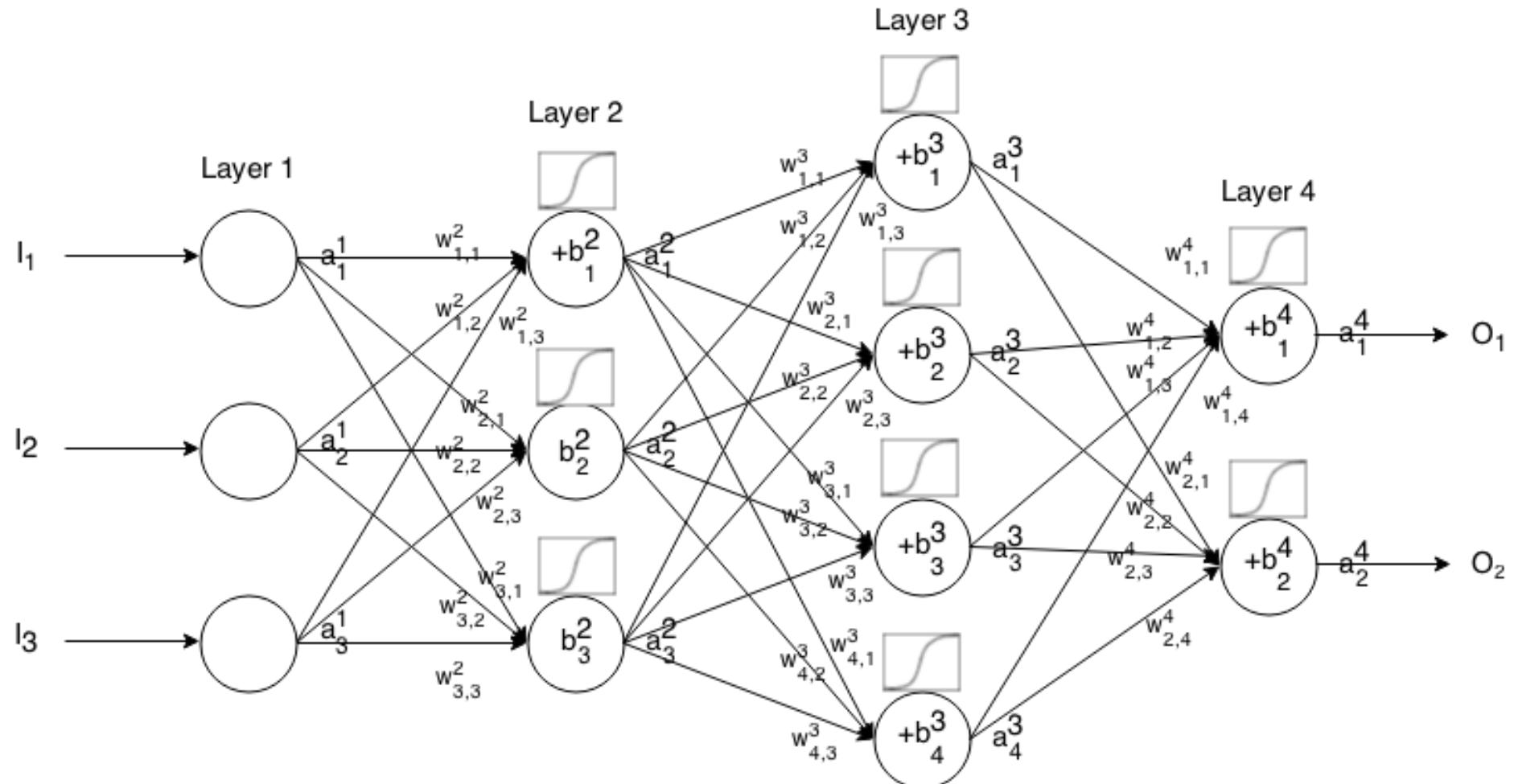


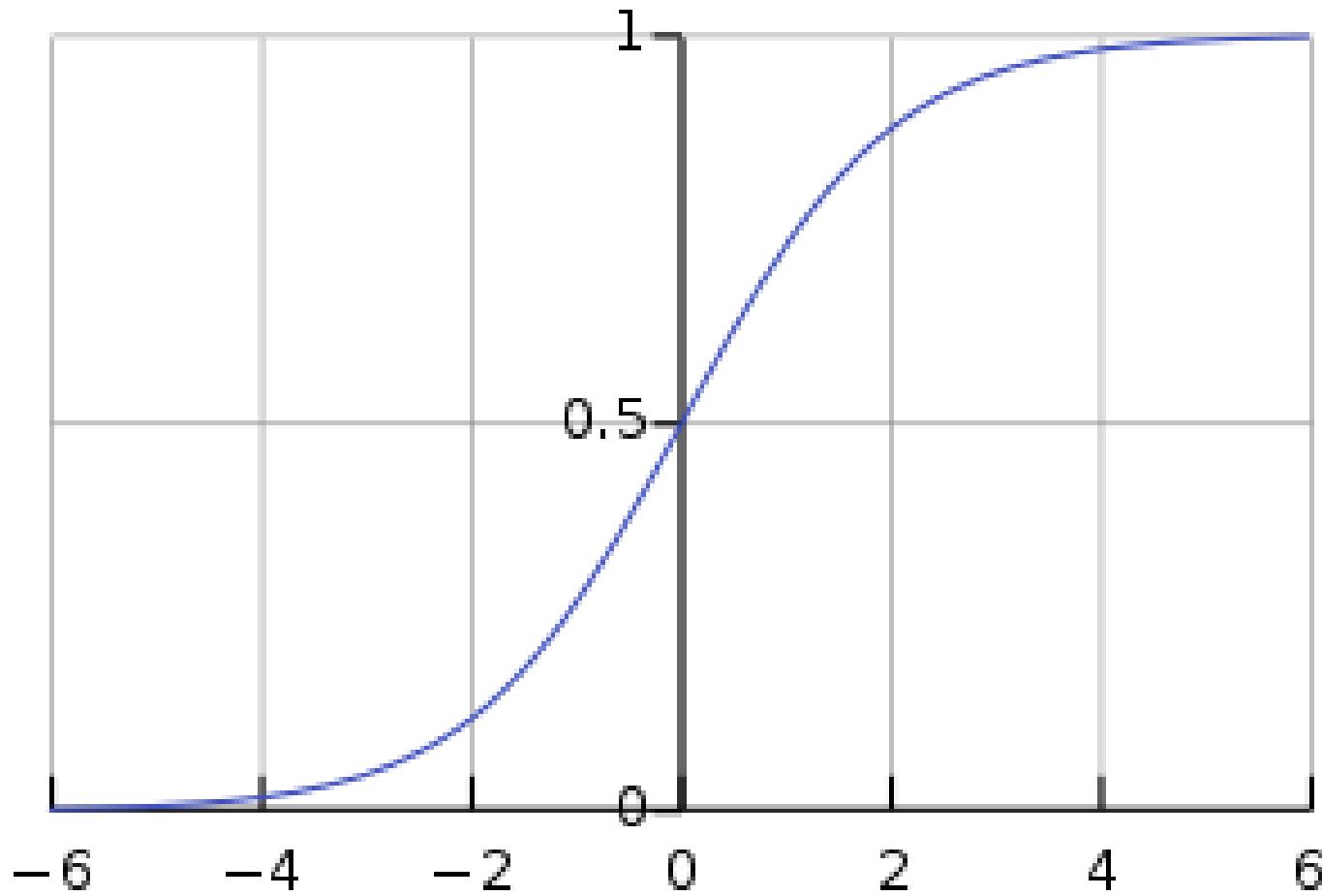
- “Deep” Learning = Neural Networks
- An old technology, but seen differently now
- Universal function approximators (flexible prediction)
- Less emphasis on feature extraction
- Got seriously popular after 2012 due to data+compute explosion
- Particularly good for vision, speech, RL and NLP due to flexible prediction architecture
- Convergence of ML and software development

# WHAT ARE NEURAL NETWORKS?



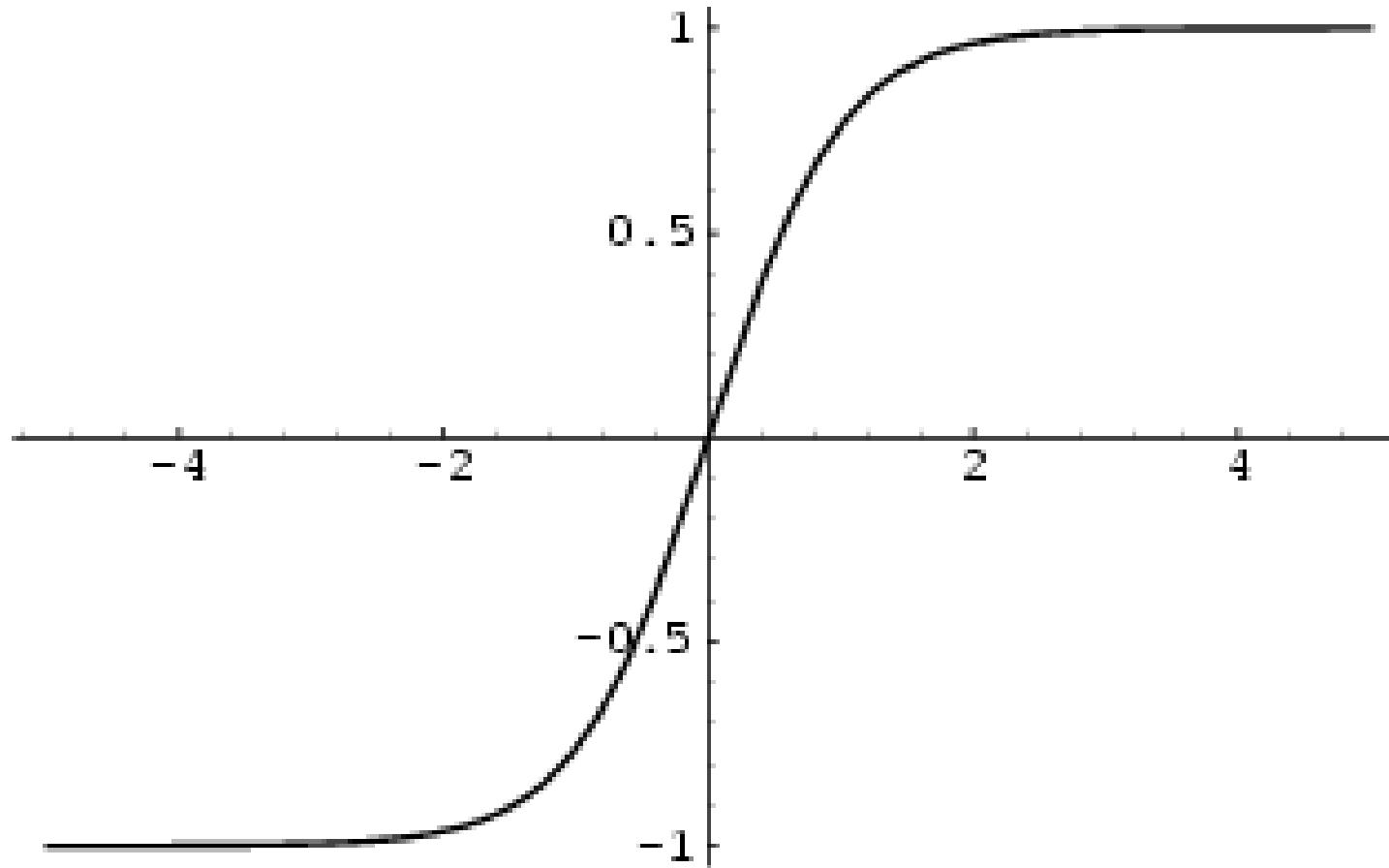
# WHAT ABOUT “DEEP” NEURAL NETWORKS?





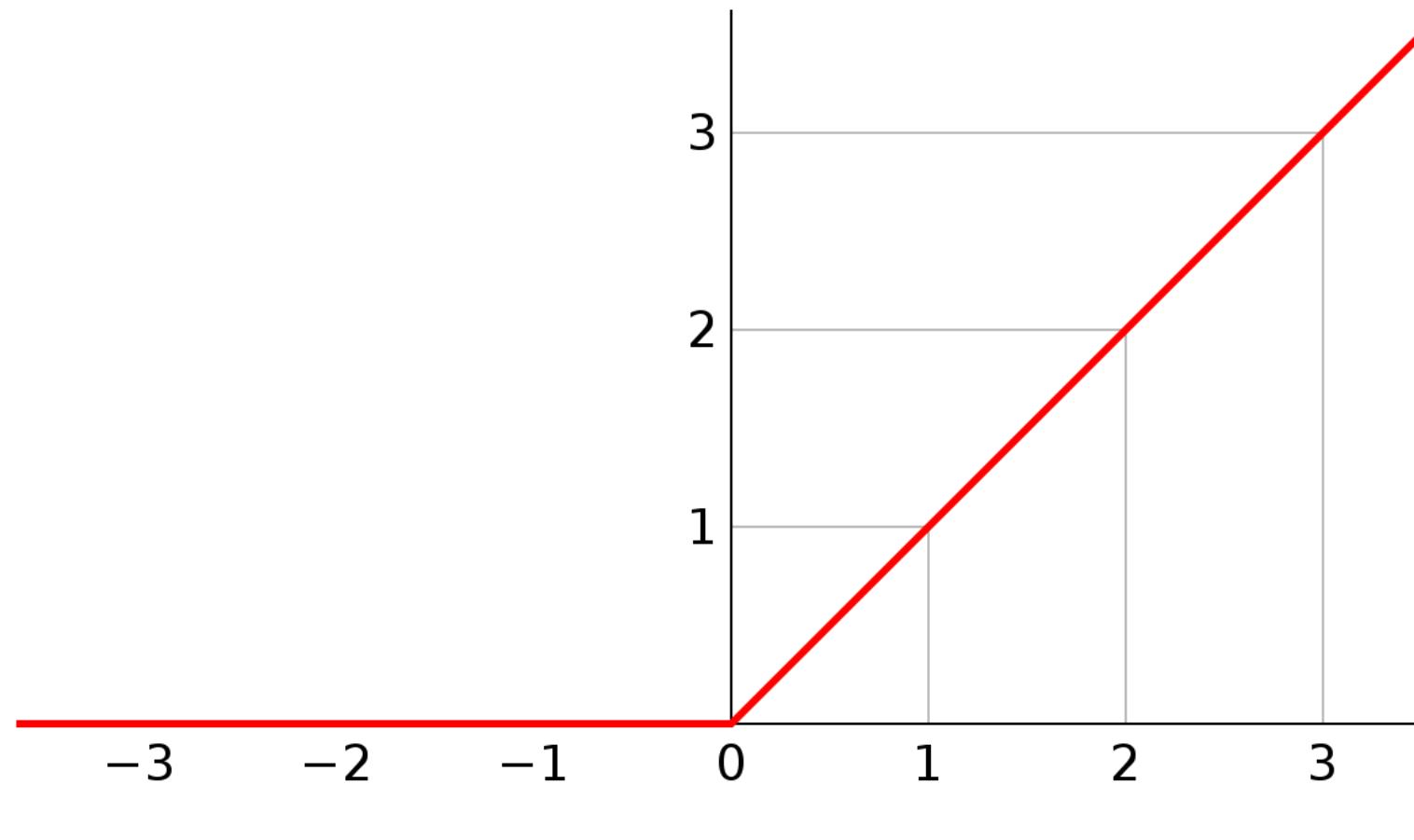
$$\frac{e^x}{e^x + 1}$$

# SIGMOID SQUASHING FUNCTION

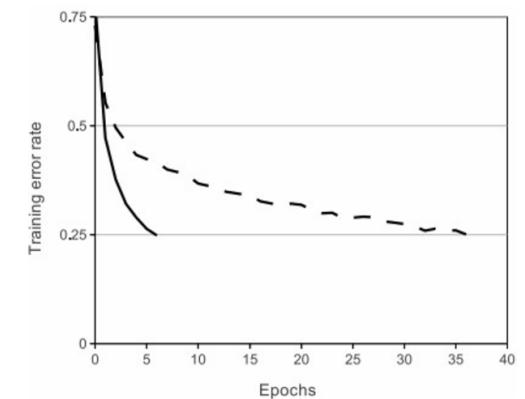


$$\frac{1 - e^{-2x}}{1 + e^{-2x}}$$

# TANH SQUASHING FUNCTION

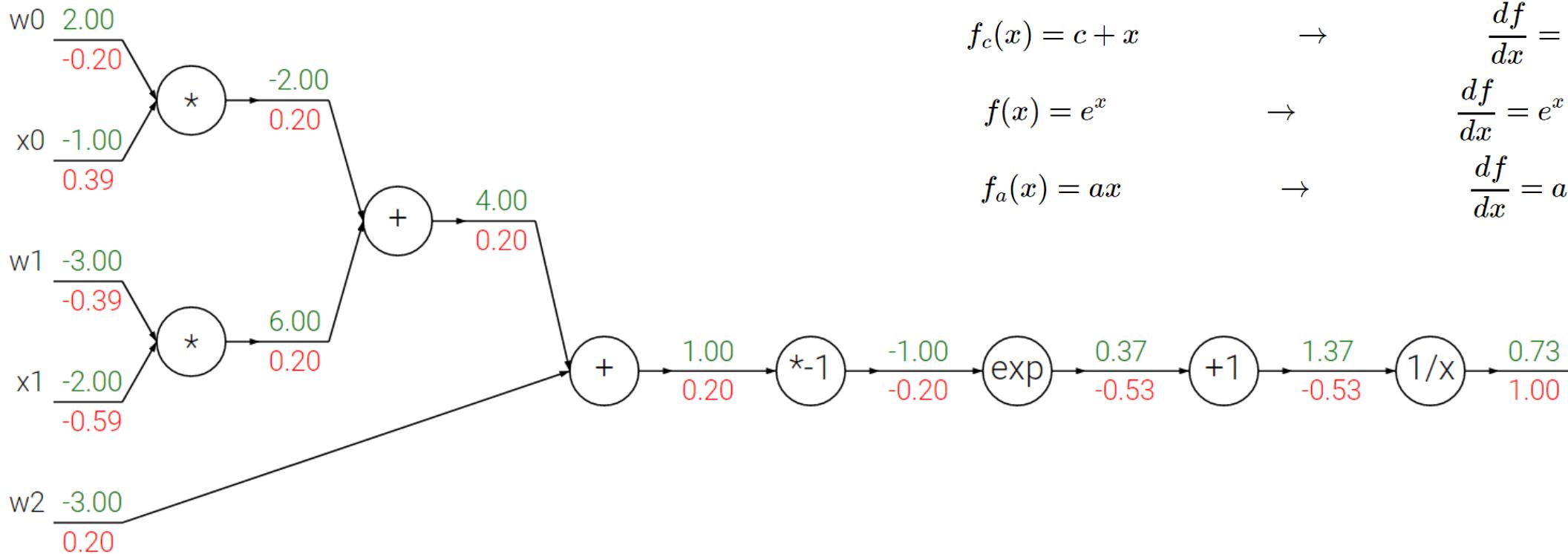


$$f(x) = x^+ = \max(0, x)$$



# RELU SQUASHING FUNCTION

# BACKPROPAGATION



Example circuit for a 2D neuron with a sigmoid activation function. The inputs are  $[x_0, x_1]$  and the (learnable) weights of the neuron are  $[w_0, w_1, w_2]$ . As we will see later, the neuron computes a dot product with the input and then its activation is softly squashed by the sigmoid function to be in range from 0 to 1.

$$\frac{\partial E}{\partial w_{jk}}$$

$$\Delta w_{jk} = \eta * [x_j * (o_k - t_k) * o_k * (1 - o_k)]$$

learning  
rate

error  $e_k$

derivative of output activation  $\varphi'_k$

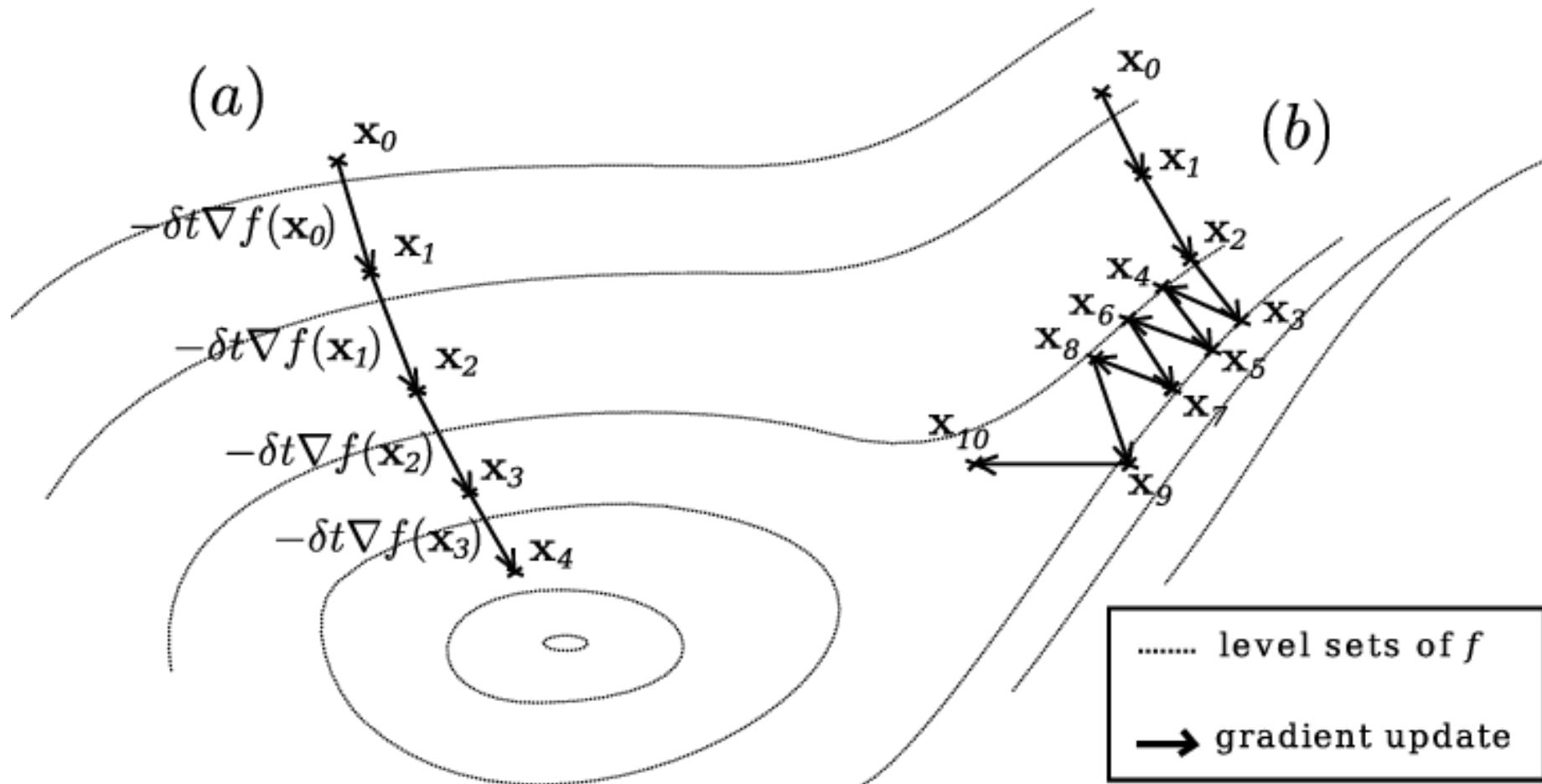
signal  $\delta_k$

# WEIGHT UPDATE

```
loop maxEpochs times
    for-each training item
        get target values
        compute output values
        compute the gradient of each weight
        use gradient to compute delta for each weight
        update each weight using its delta
    end-for
end-loop
```

# BACKPROP ALGORITHM

# OPTIMIZATION/GRADIENT DESCENT



WEIGHT UPDATES IN EPOCHS / TRAINING DATA SPLIT INTO MINIBATCHES

DEMO

# NEURAL NETWORK PLAYGROUND



DEMO

<https://github.com/ecsplendid/deep-learning-cntk-talk>

# CNTK IRIS DEMO



# WHAT IS CNTK?

DECLARITIVELY DESCRIBE AND TRAIN DEEP NEURAL NETWORKS

DOES ALL THE HARD WORK FOR YOU

80% INTERNAL MS DL WORKLOADS USE CNTK

1<sup>ST</sup> CLASS ON LINUX, WINDOWS, DOCKER

BRAINSRIPT, DON'T NEED TO KNOW HOW TO CODE

C#, PYTHON, R, COMMANDLINE

KERAS BINDINGS

V2.2 JUST RELEASED

OPERATIONALISE WITH THE AZUREML-CLI OR AZURE FUNCTIONS NOW THAT WE SUPPORT C#?!



<http://dlbench.comp.hkbu.edu.hk/>

Benchmarking by HKBU, Version 8

Single Tesla K80 GPU, CUDA: 8.0 CUDNN: v5.1

Caffe: 1.0rc5(39f28e4)

CNTK: 2.0 Beta10(1ae666d)

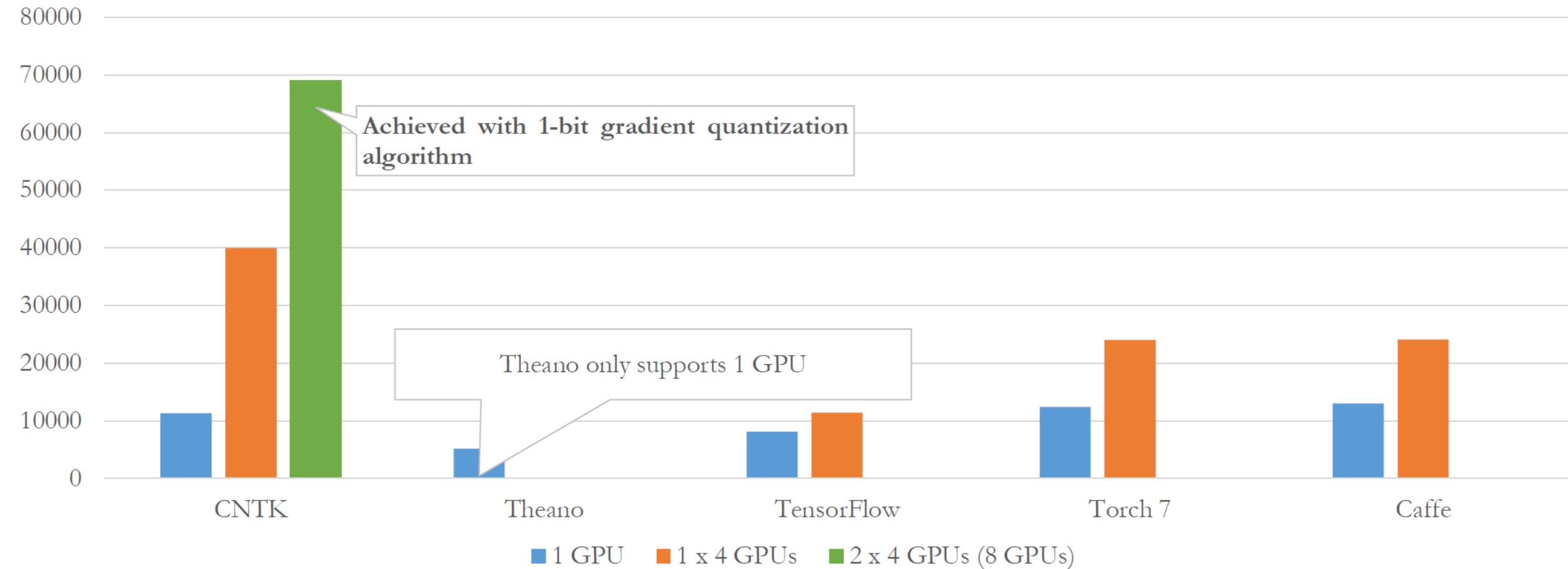
MXNet: 0.93(32dc3a2)

TensorFlow: 1.0(4ac9c09)

Torch: 7(748f5e3)

	Caffe	CNTK	MxNet	TensorFlow	Torch
FCN5 (1024)	55.329ms	<b>51.038ms</b>	60.448ms	62.044ms	52.154ms
AlexNet (256)	36.815ms	<b>27.215ms</b>	28.994ms	103.960ms	37.462ms
ResNet (32)	143.987ms	<b>81.470ms</b>	84.545ms	181.404ms	90.935ms
LSTM (256) (v7 benchmark)	-	<b>43.581ms</b> (44.917ms)	288.142ms (284.898ms)	- (223.547ms)	1130.606ms (906.958ms)

# THE FASTEST TOOLKIT



# MOST SCALABLE TOOLKIT (2016)

# INSTALLING CNTK

- GOOGLE “CNTK INSTALL” (WITH BING)
- USE THE “SCRIPT DRIVEN INSTALLATION”

# WHEN TO USE DEEP LEARNING FRAMEWORKS

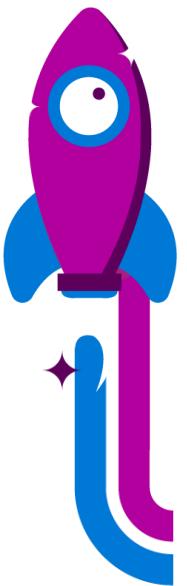


- Sequence modelling (speech, language, time-series)
- Complex vision tasks (localisation, detection)
- Novel prediction architectures
- Generative models
- Reinforcement learning
- ... and many more!

# DEEP LEARNING ON AZURE CLOUD



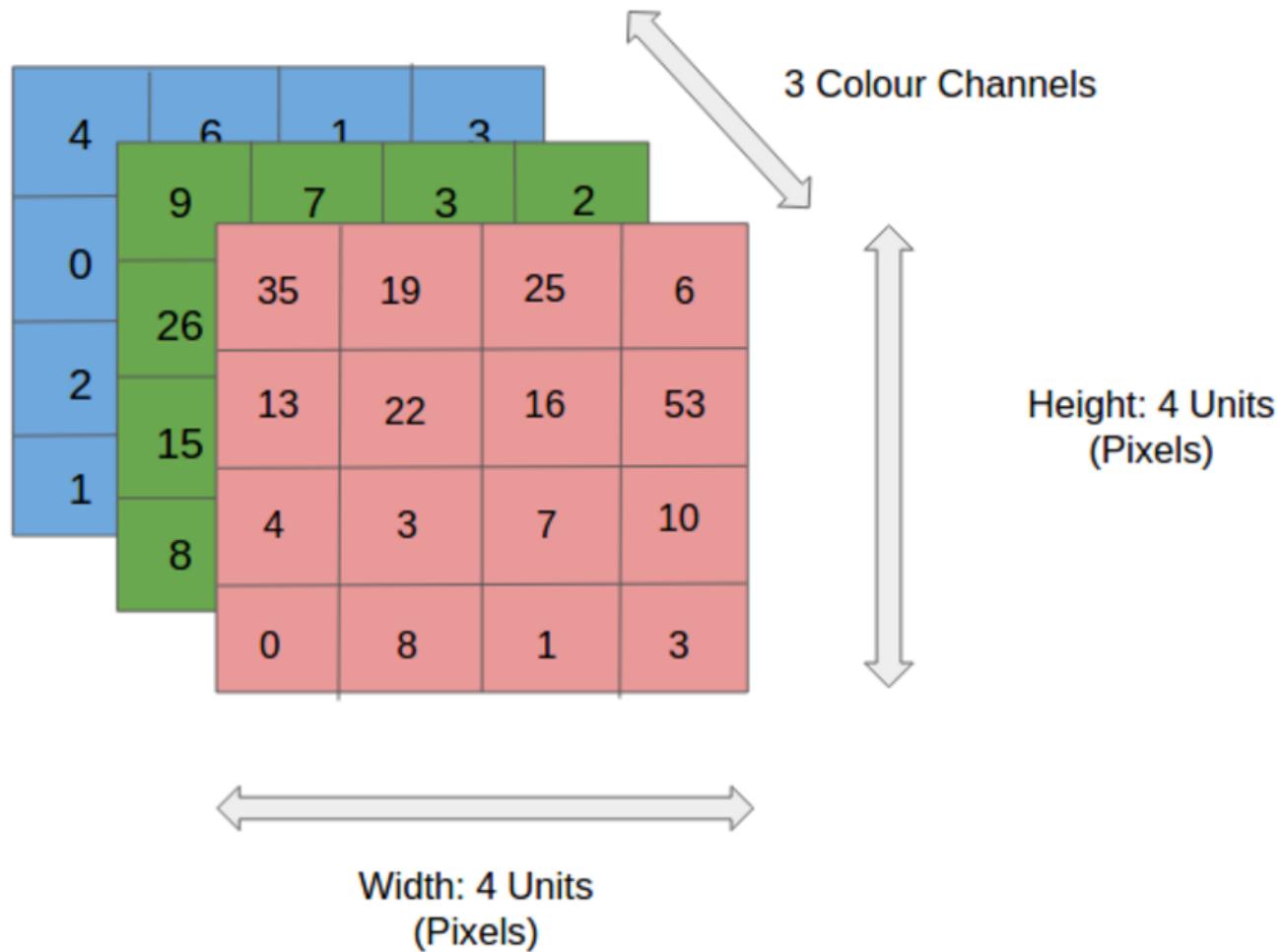
- Data Science Virtual Machine (Ubuntu and Windows)
- Batch AI Training Service
- FPGA/ASICs are coming soon
- AzureML supports discriminative neural networks i.e. regression and classification
- R Server includes a trained deep learning image featurizer



# How do convolutional neural networks work?



# PREPARE DATASET OF IMAGES



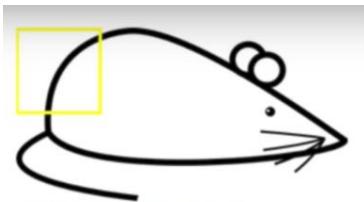
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

# CONVOLUTION FILTER



Visualization of the filter on the image



Visualization of the receptive field

$$(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$$

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

\*

0	0	0	0	0	0	30	0
0	0	0	0	0	30	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	30	0	0	0	0
0	0	0	0	0	0	0	0

Pixel representation of filter

# CONVOLUTION FILTER MATCH



Visualization of the filter on the image

MULTIPLY AND SUMMATION = 0

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

\*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

CONVOLUTION FILTER NO MATCH

DEMO

# CONVOLUTION FILTERING IN MATLAB

1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	1 x1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

1	1	1	0	0
0	1	1 x1	1 x0	0 x1
0	0	1 x0	1 x1	1 x0
0	0	1 x1	1 x0	0 x1
0	1	1	0	0

Image

4	3	4
2	4	3

Convolved Feature

1	1	1	0	0
0	1	1	1	0
0	0	1 x1	1 x0	1 x1
0	0	1 x0	1 x1	0 x0
0	1	1 x1	0 x0	0 x1

Image

4	3	4
2	4	3
2	3	4

Convolved Feature

# CONVOLUTION



4	6	1	3
0	8	12	9
2	3	16	100
1	46	74	27



35	19	25	6
13	22	16	63
4	3	7	10
9	8	1	3



9	7	3	2
26	37	14	1
15	29	16	0
8	6	54	2



35	19	25	6
13	22	16	63
4	3	7	10
9	8	1	3



(i)

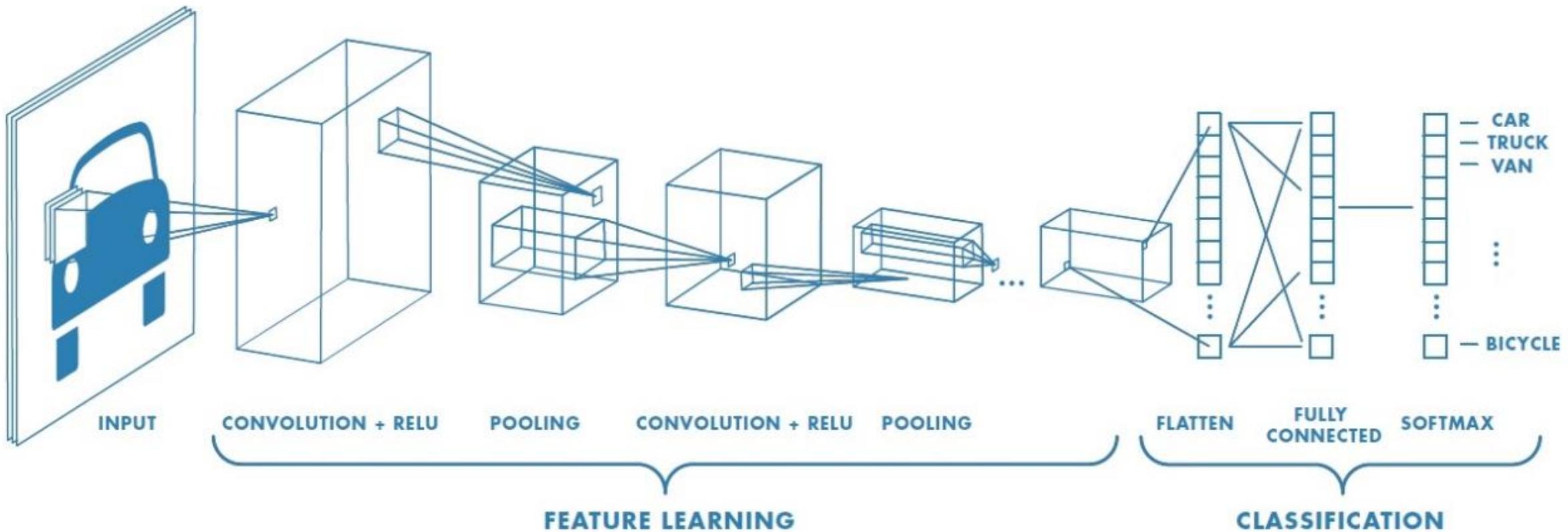
(iii)

(ii)

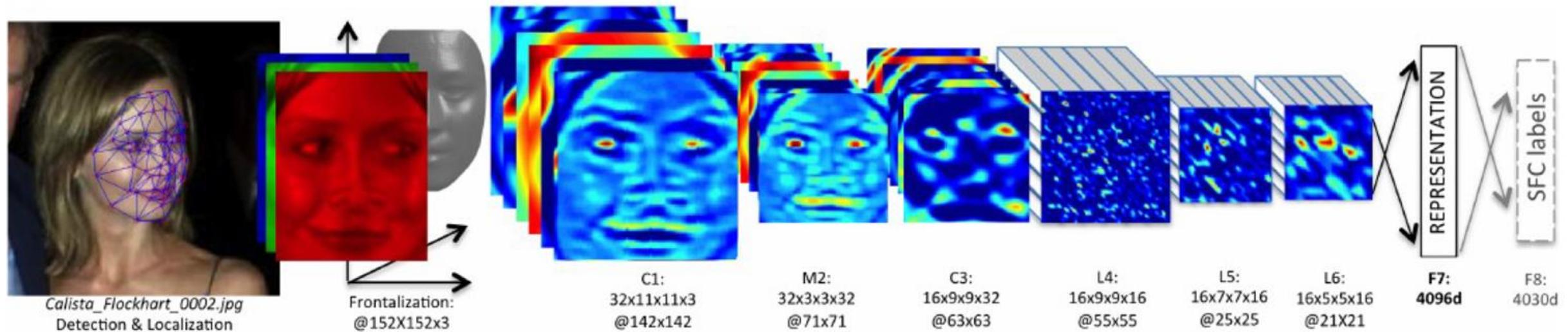
(iv)

# POOLING

# Image Classification Example

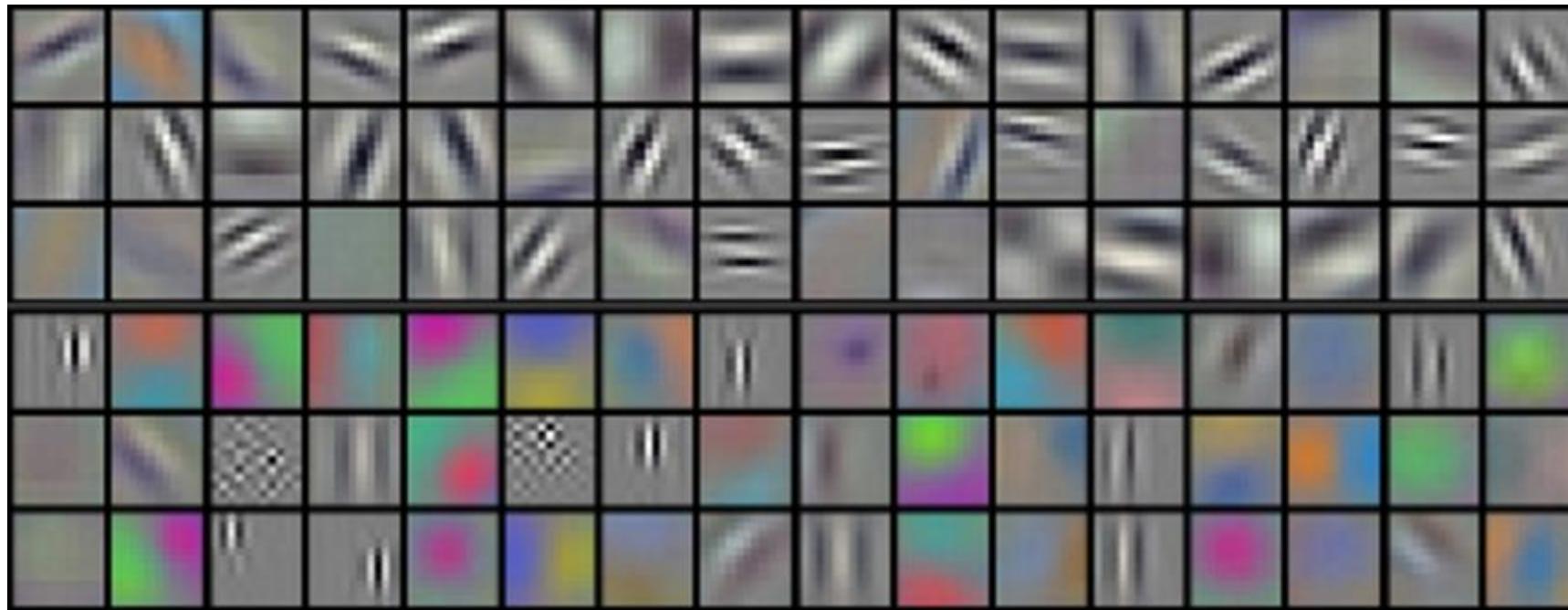


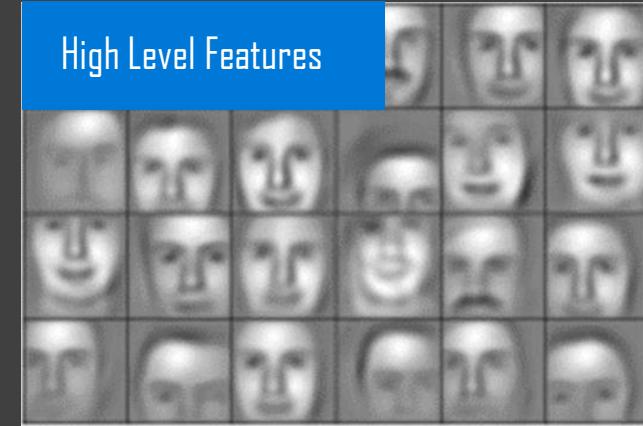
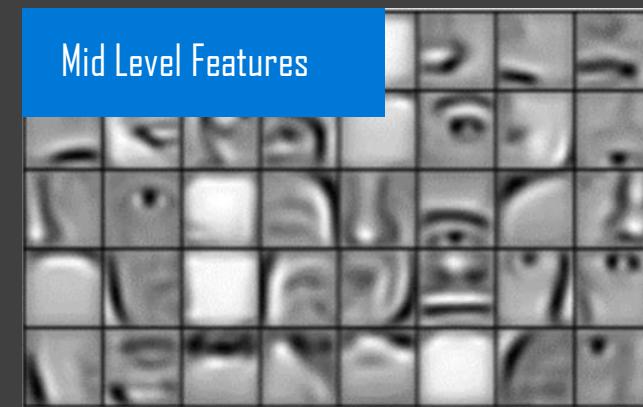
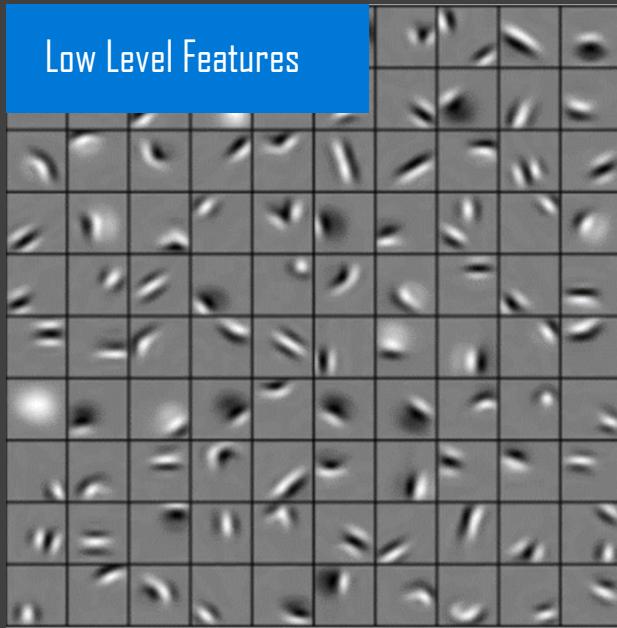
# DEEP FACE



- Alignment
- CNN
- Metric Learning

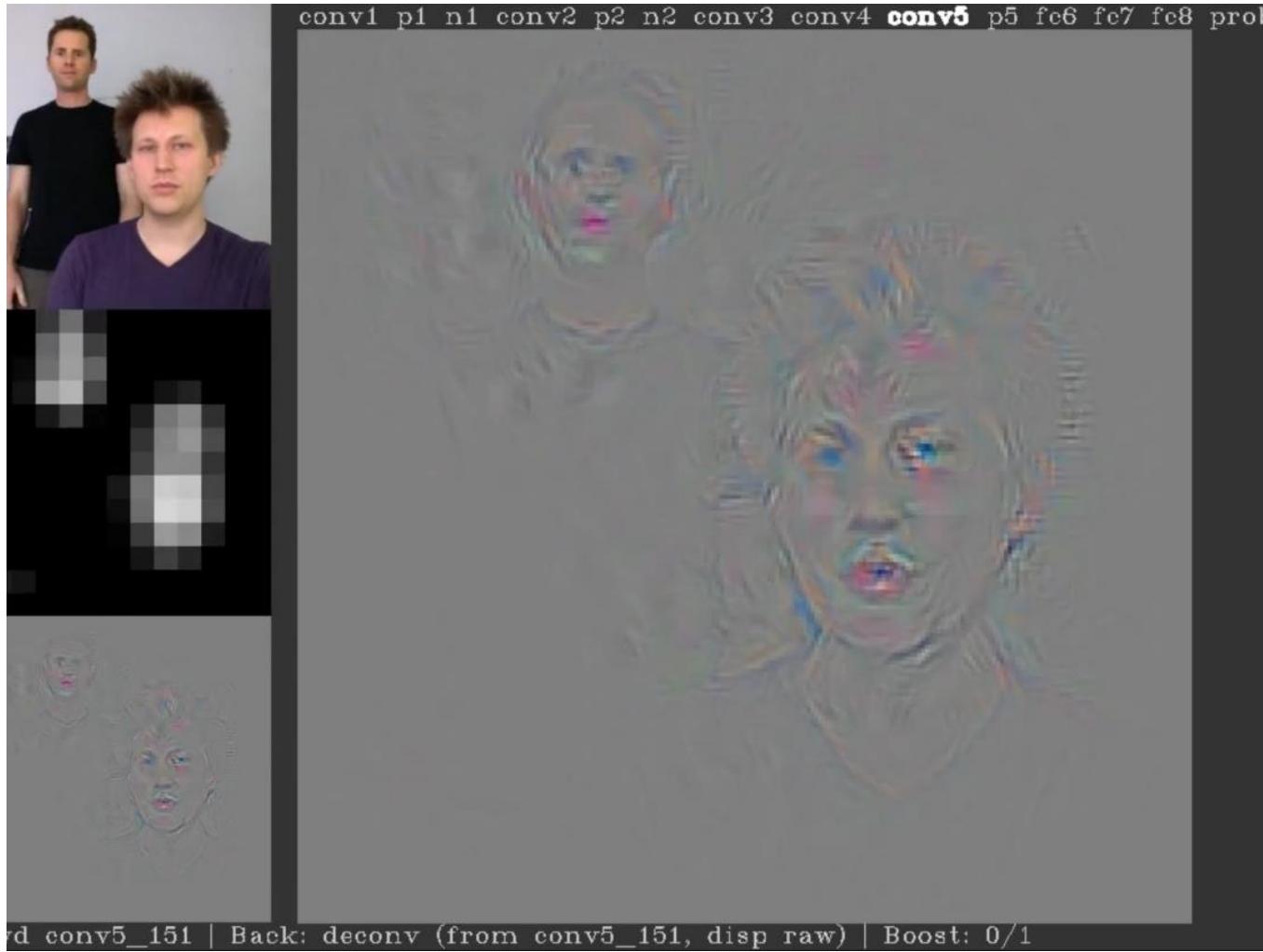
# VISUALISING THE FILTERS





**IF IT'S "DEEP" - IT HAS MORE THAN ONE LAYER OF FEATURE TRANSFORMATION**

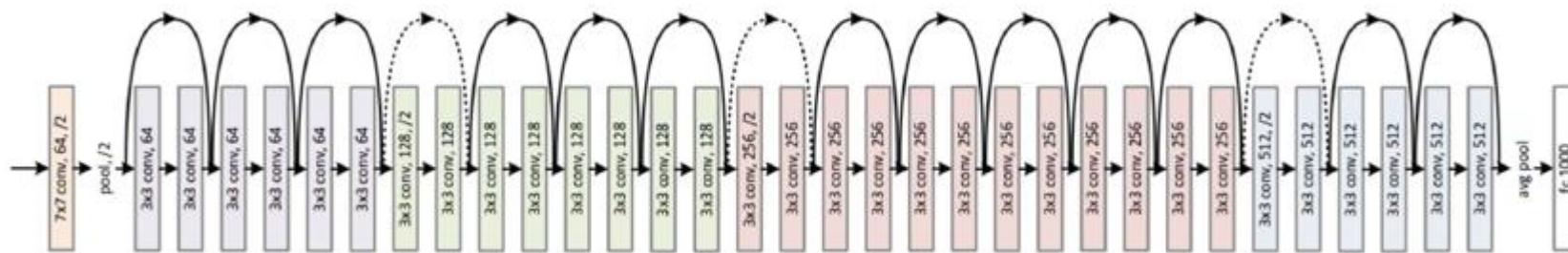
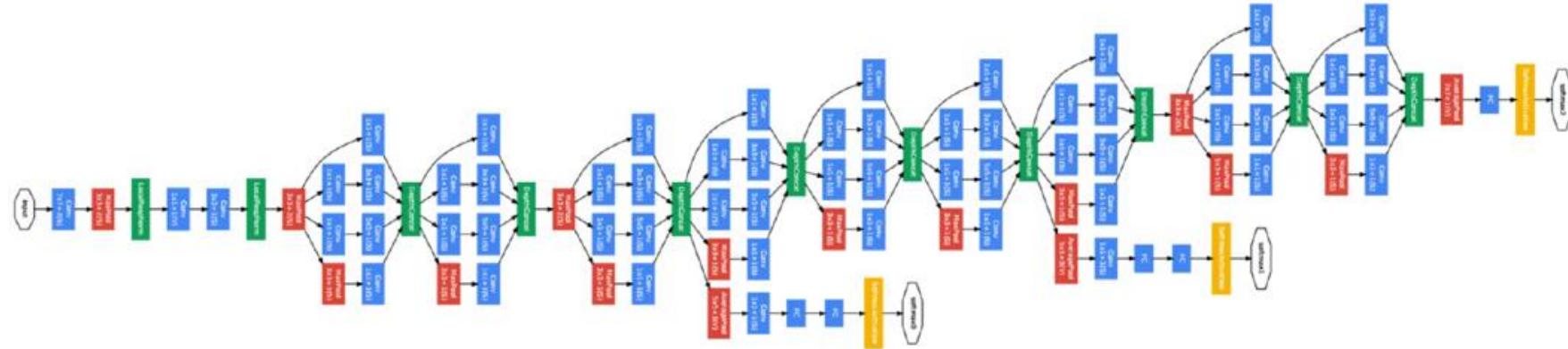
# DEEP VISUALISATION TOOLBOX



<https://www.youtube.com/watch?v=AgkfIQ4lGaM>



VGGNet, Oxford Uni, 2014

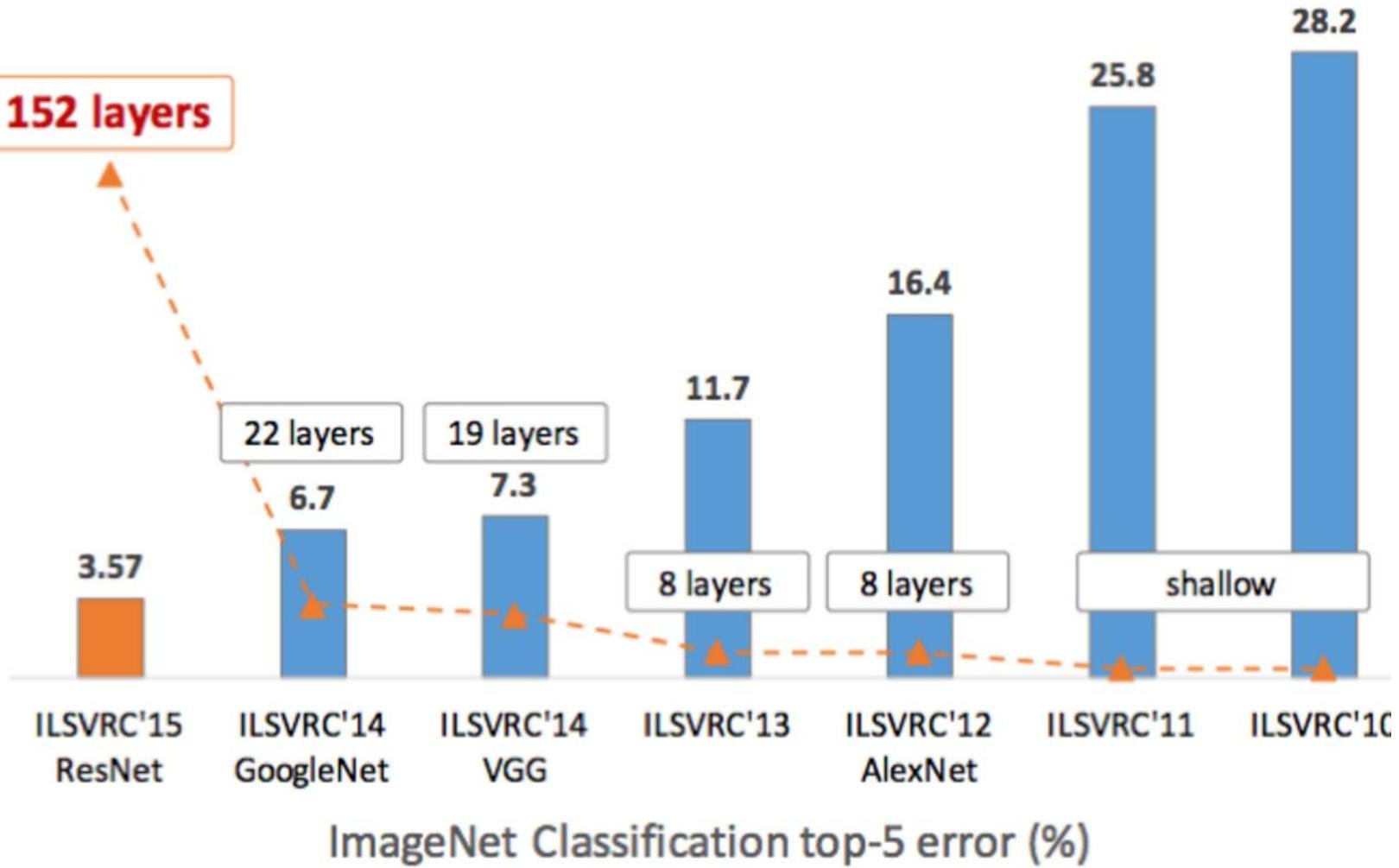


GoogLeNet, Google, 2015

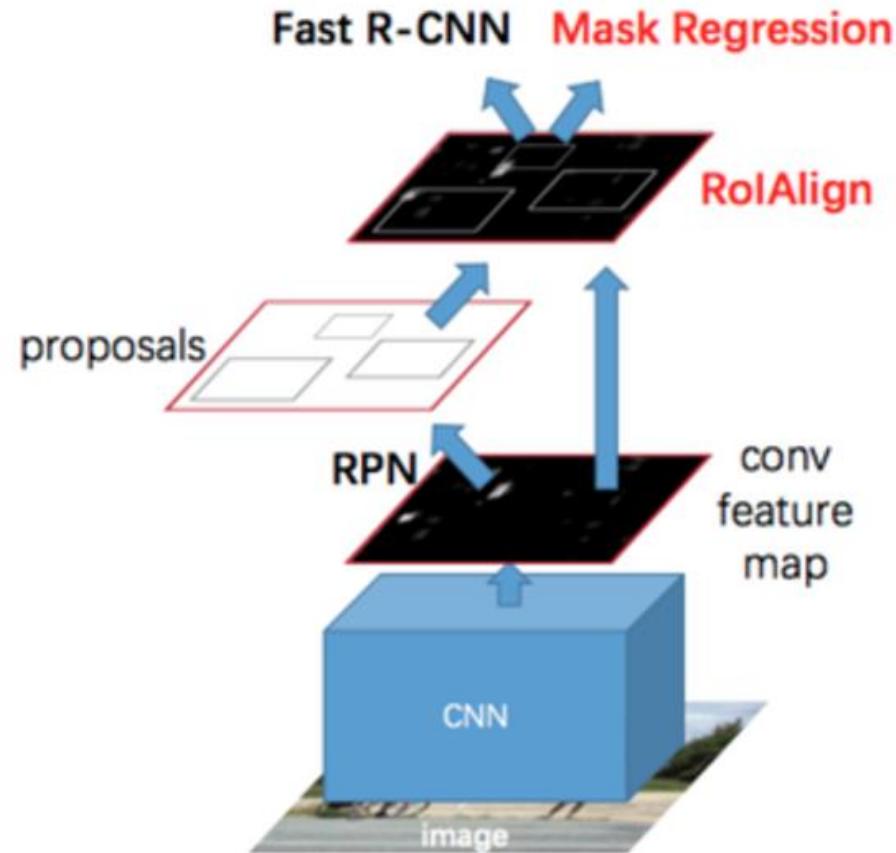
ResNet, Microsoft, 2016

# WINNING ARCHITECTURES

# RESOLUTION OF DEPTH

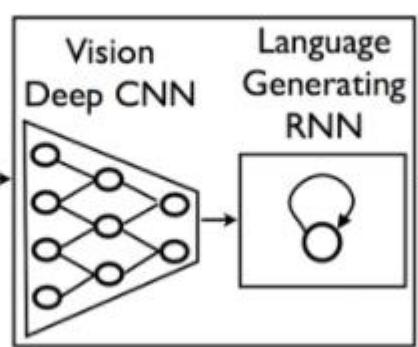


# MASK R-CNN ARCHITECTURE (2017)

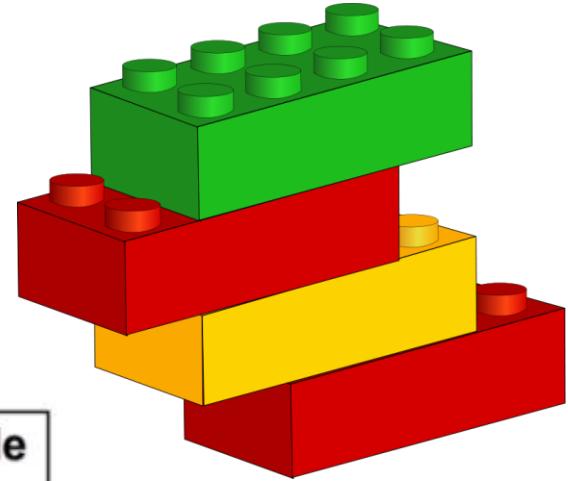


## ARCHITECTURE COMPOSABILITY

# BUILDING PREDICTIVE ARCHITECTURES LIKE LEGO BLOCKS



**A group of people shopping at an outdoor market.**  
**There are many vegetables at the fruit stand.**



(Vinyals, et al. (2014))



# COUNTING SHEEP

# IS DEEP LEARNING A FAD?

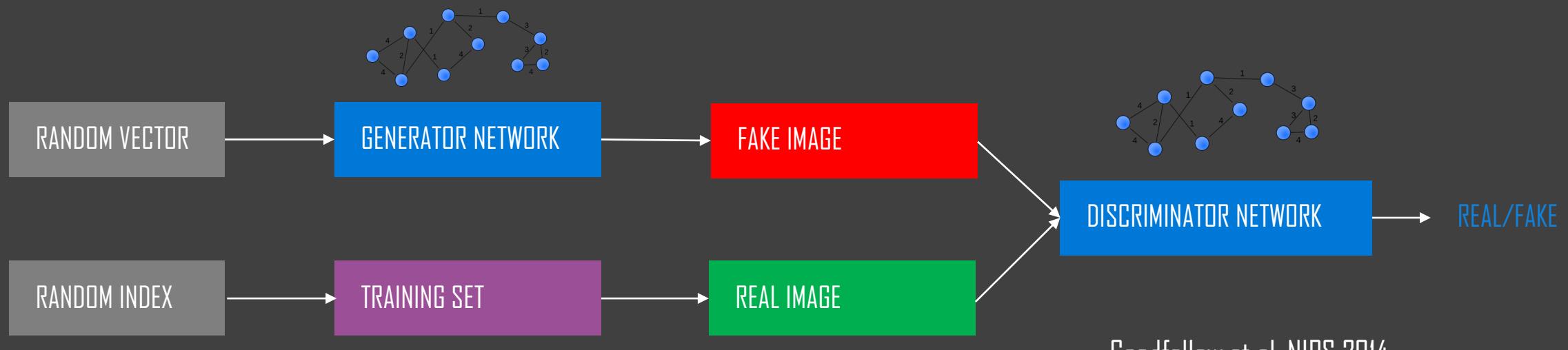


- The way we think about neural networks now is totally different to 30 years ago
- Previous frequentist algorithms were just learning weighted combinations of hand-crafted features
- NNs learn a hierarchy of representations which work really well in many domains
- Before we used to talk about classification and regression, now we talk about *predictive architectures, many of which are game-changing.*

# TWO INNOVATIVE DL ARCHITECTURES



- An example; generative adversarial networks
- Game-theoretical approach to deep learning



Goodfellow et al. NIPS 2014



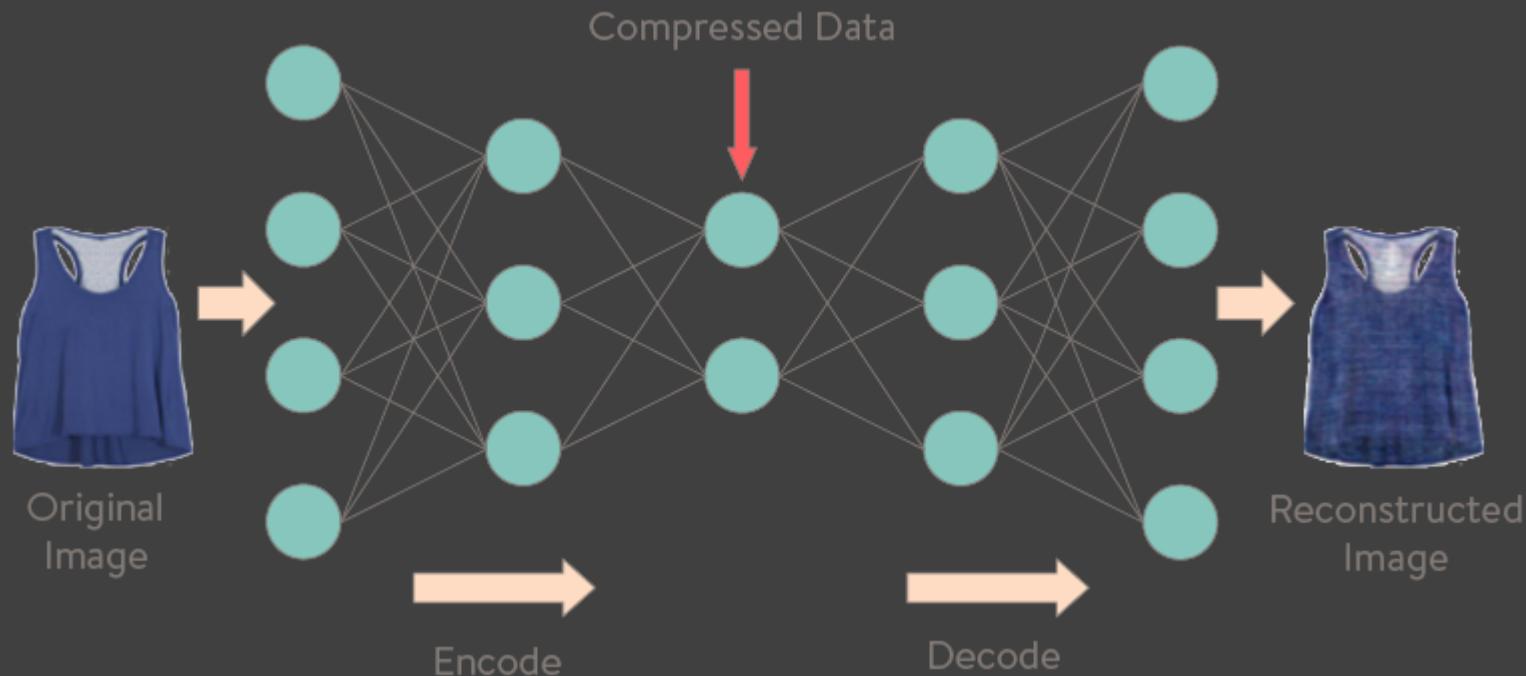
## VERY ADVANCED NOVEL ARCHITECTURES (GANS)



# IMAGES GENERATED WITH GANS



- Autoencoders are an unsupervised method to compress information into an “embedding” similar to unsupervised dimensionality reduction.



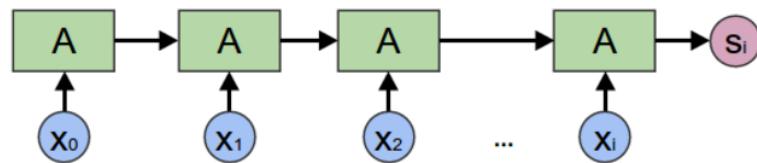
# AUTOENCODERS

# PARADGM-SHIFT

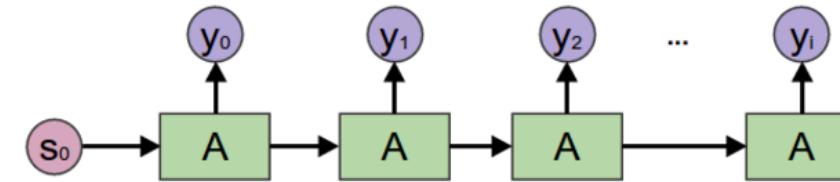


- Three narratives currently exist to describe deep learning
  - Neuroscience
  - Probabilistic
  - Manifold
- The *differentiable programming* narrative is emerging

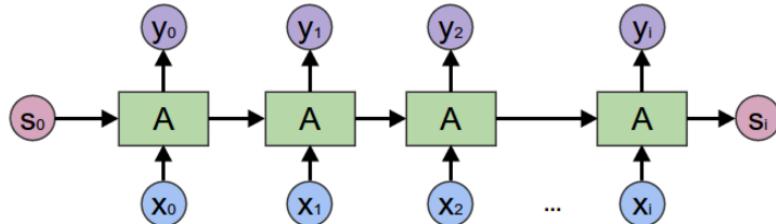
# FUNCTIONAL PROGRAMMING IN DL



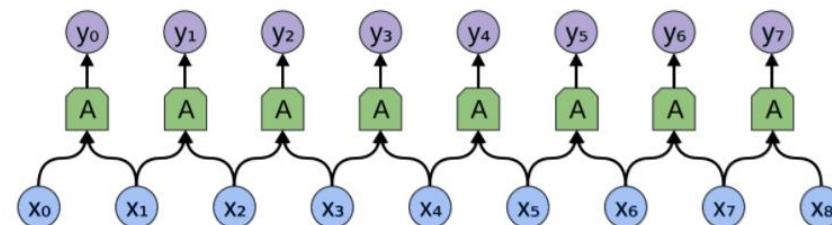
**fold = Encoding RNN**  
Haskell: `foldl a s`



**unfold = Generating RNN**  
Haskell: `unfoldr a s`



**Accumulating Map = RNN**  
Haskell: `mapAccumR a s`



**Windowed Map = Convolutional Layer**  
Haskell: `zipWith a xs (tail xs)`



# LEARNING MORE: AZUREWIKI, ML DISTRIBUTION LIST

machine-learning@microsoft.com

The screenshot shows a Microsoft SharePoint-like interface titled "AzureWiki". The top navigation bar includes tabs for "Squad Information", "AWS Compete", "BCDR", "Conversation as a Service", "Docker", "Data Migration Squad", "Machine Learning" (which is highlighted in pink), "Azure AD App Proxy", "Micro Services", and "OSS". A sidebar on the right contains links such as "Add Page", "Intro to Machine Learning", "Learn Deep Learning!", "Learn Machine Learning!", "Microsoft ML stack", "MS Roadmap", "CNTK", "Installing CNTK", "101 Deck & Docs", "Contacts (SMEs)", "Project Vienna", and "DSVM".

**Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM)**  
<https://www.youtube.com/watch?v=WCUNPb-5EYI>

How Convolutional Neural Networks work  
<https://www.youtube.com/watch?v=FmpDlaiMleA>

How data science works  
<https://www.youtube.com/watch?v=tKa0zDDDaQk>

How Bayes Theorem works (not deep learning)  
<https://www.youtube.com/watch?v=5NMxiOGL39M>

**CS231n Convolutional Neural Networks for Visual Recognition**  
This course from Stanford with Andrej Karpathy is probably the best (comprehensive) learning resource I have found on Deep Learning and CNNs. If you were only going to look at one thing to get a comprehensive background, it would be this.

Check the website here <http://cs231n.github.io>

This website is literally a gold mine, I suggest studying it page for page. It has a great run down of Python for machine learning and detailed discussion of NNs and CNNs.

The lectures for this module are also available on YouTube;  
<https://www.youtube.com/watch?v=NfnWJuUJYU&list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AIC>

<update 16 Aug 2017>  
Couple of days ago Stanford posted a new set of lectures for this course from Spring 2017! So if you are just starting out, maybe watch these ones instead. This is now the third time they have run this course. One caveat is that Andrej Karpathy fronted the first set of videos and he's really amazing (check his website <http://karpathy.github.io>).  
<https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv>

The lectures detail NNs, Optimisation, Backprop, Convnets, Localisation/Detection, RNNs/LSTMs, segmentation etc

**CS224n: Natural Language Processing with Deep Learning**  
Probably the best course on understanding NLP with respect to deep learning. Slightly more focus on mathematical derivation compared to the 231n course.

# THANK YOU

[tim.scarfe@microsoft.com](mailto:tim.scarfe@microsoft.com)

