

Deep Learning on Azure

Dr. Tim Scarfe 

Data Solution Architect / Data Scientist

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





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

Satya Nadella

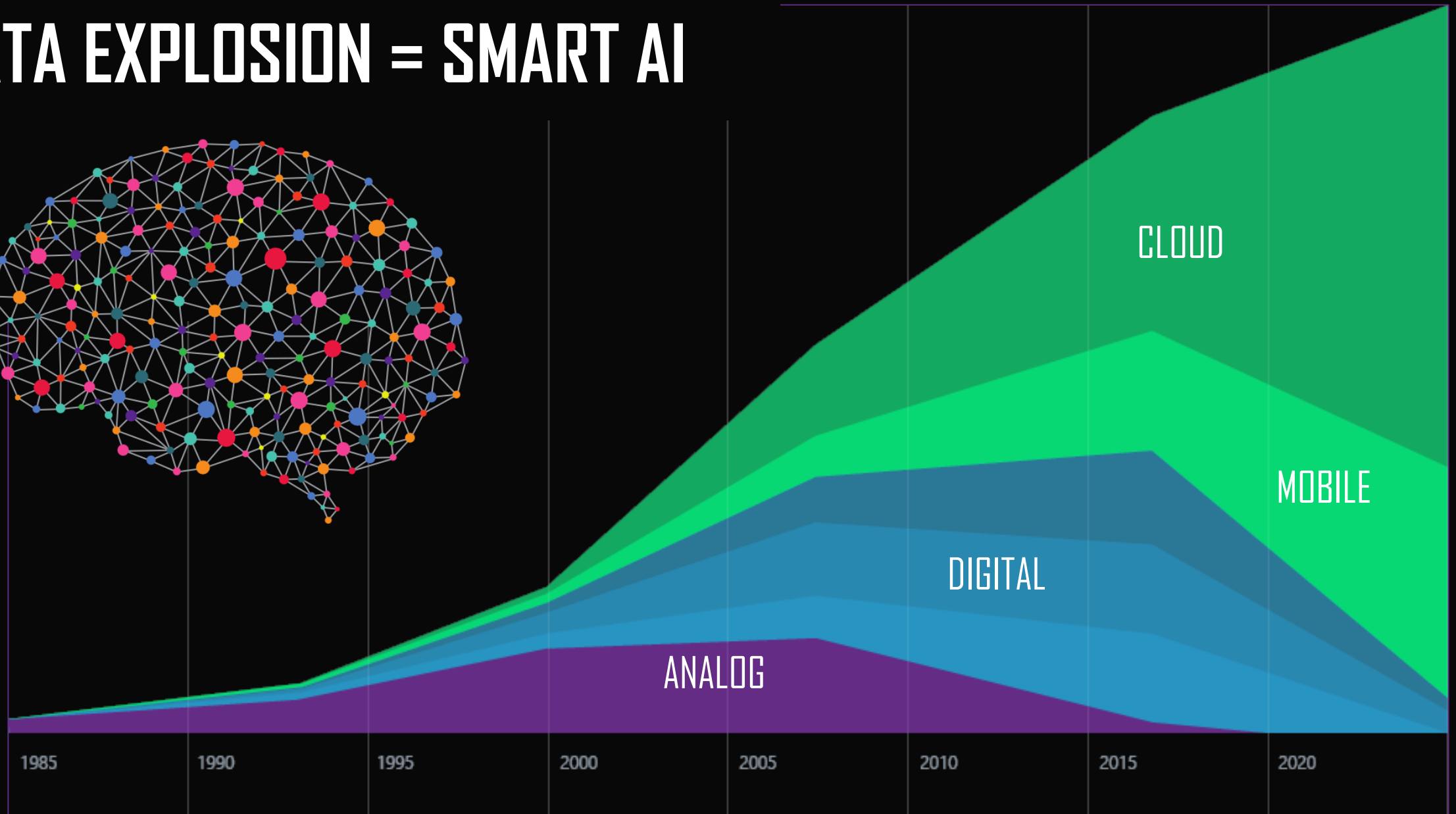


WHAT IS MACHINE LEARNING?

Machine learning is about learning from previous experience so you can make accurate predictions about the future.



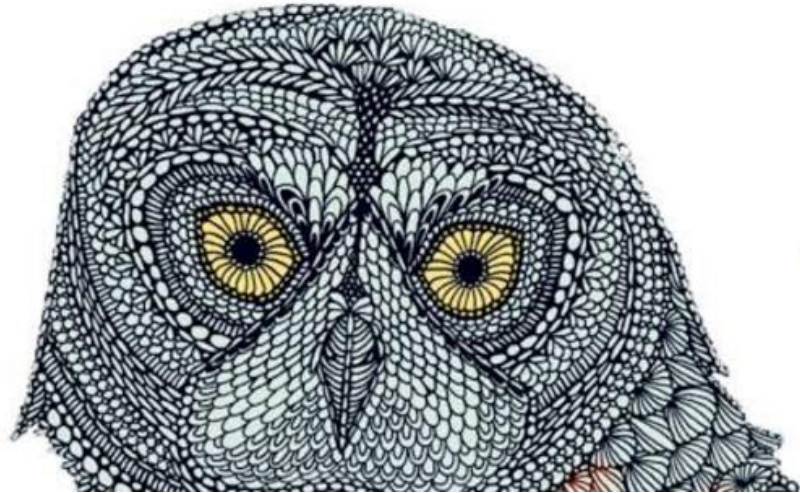
DATA EXPLOSION = SMART AI



NICK BOSTROM

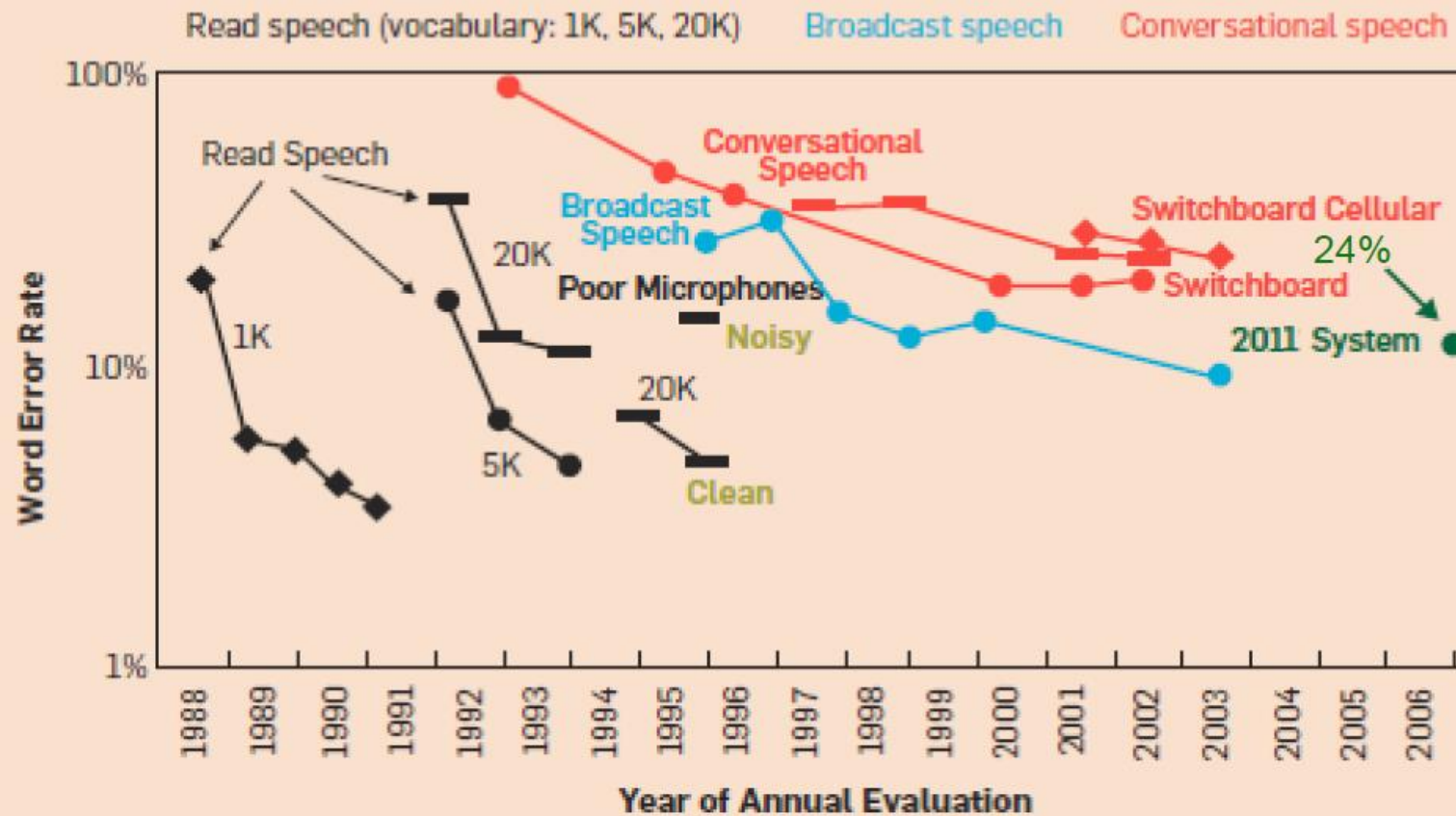
SUPERINTELLIGENCE

Paths, Dangers, Strategies



DON'T WORRY ABOUT SUPERINTELLIGENCE





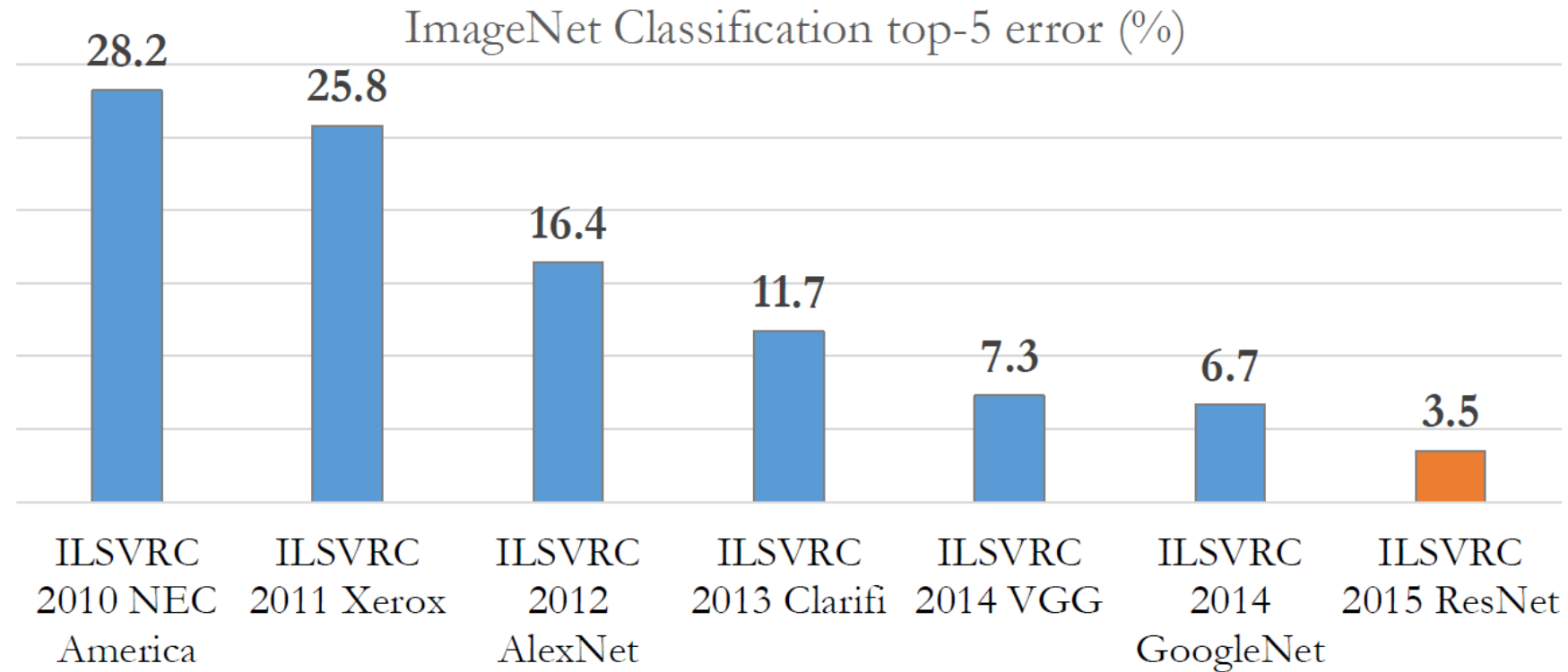
2017: ~6%!

Human error: ~5%

IMPROVEMENTS IN SPEECH RECOGNITION



IMPROVEMENTS IN COMPUTER VISION



2017: ~2.2%

Machine Learning/AI Stack at Microsoft



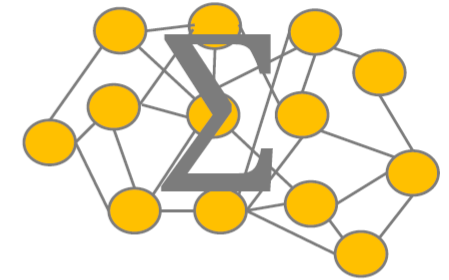
Cognitive Services



Azure
Machine Learning



R Server
Data Science
Languages



Cognitive Toolkit (CNTK)

Project "Vienna" (2018)

SaaS (REST)

PaaS (Drag/Drop)

Code
(R vs Python)

- 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
- Project "Vienna" will support all execution environments on-prem and cloud (cloud/containers/Spark)

THE NEW MICROSOFT



Batch vs. on-line

Regression

*Predicting
sequences/images*

Supervised

Anomaly Detection

Unsupervised

Agent Based Learning

Classification

Clustering

**Reinforcement
Learning**



TYPES OF MACHINE LEARNING

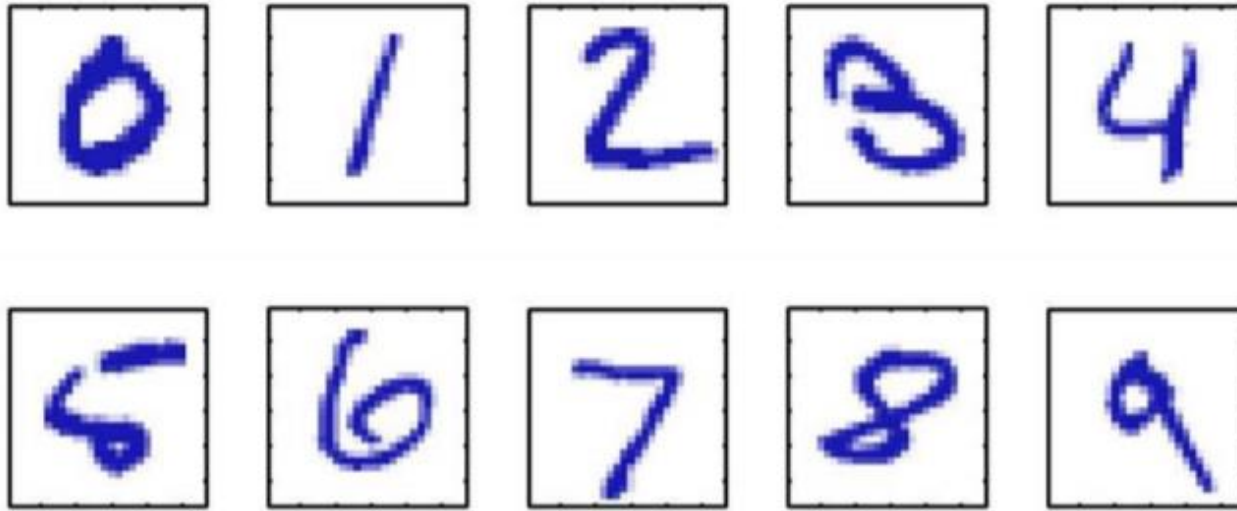
- Approximate a function which maps from signals (image) to labels (has-cat)
- This “decision function” can predict missing labels on new, previously unseen signals.
- Historically; different algorithms for different tasks
 - now; deep learning does everything



WHAT DO MACHINE LEARNING ALGORITHMS DO?



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

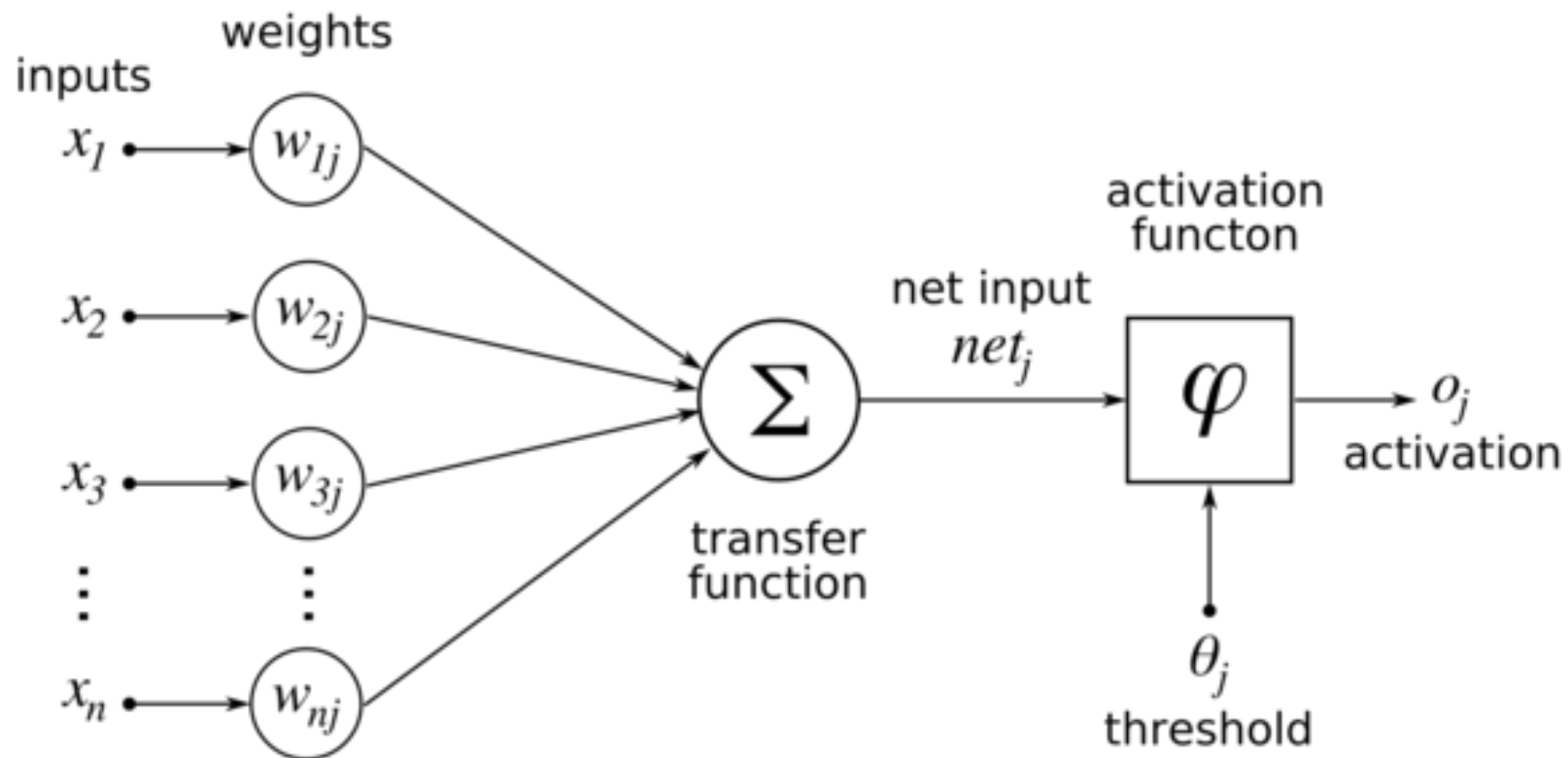
DEEP LEARNING/NEURAL NETWORK DISCUSSION



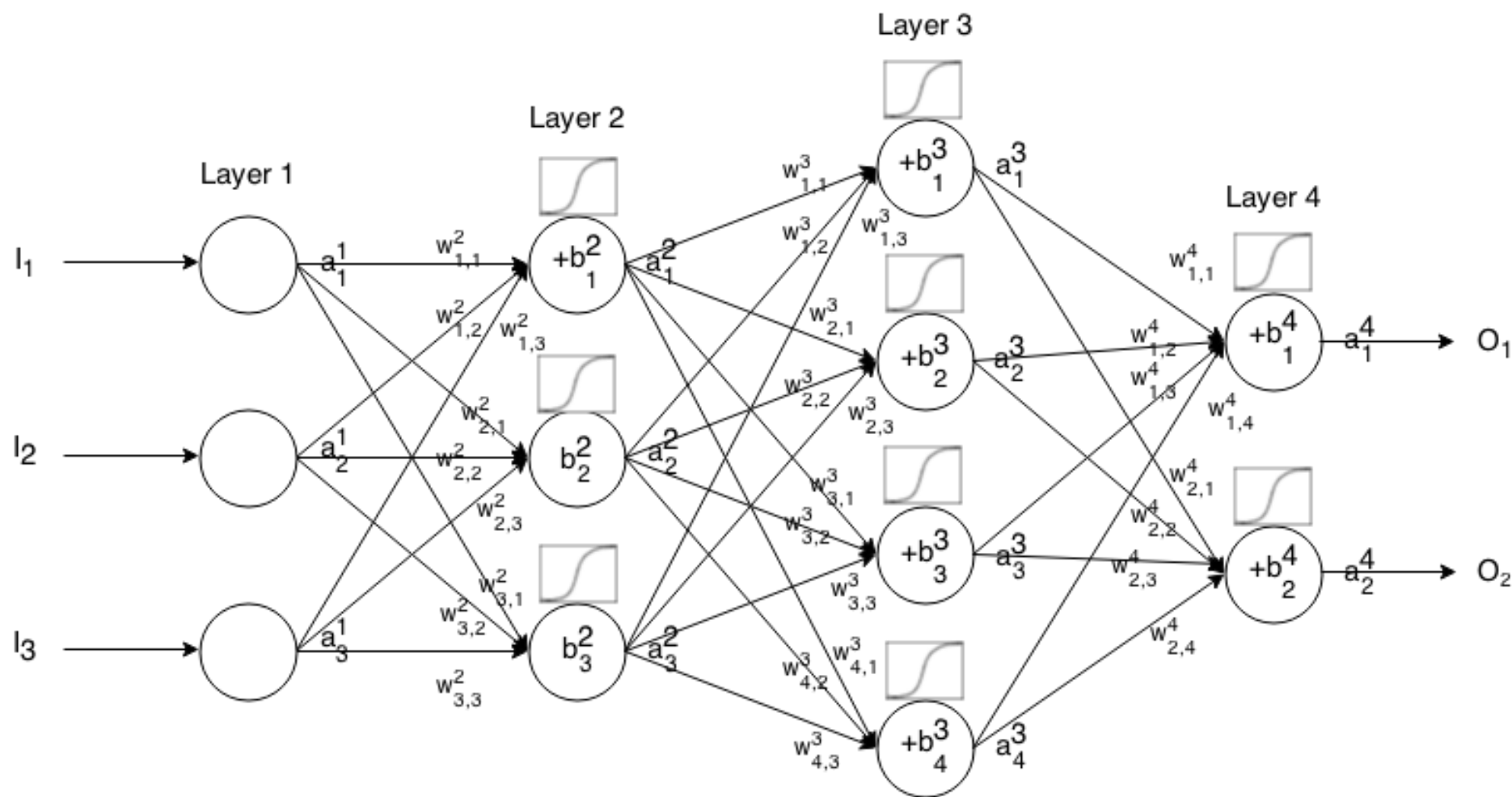
- Deep Learning = Neural Networks
- Actually, an old technology!
- Universal function approximators; extremely flexible prediction scenarios
- Less emphasis on feature extraction
- Got seriously popular after 2012 due to data+compute explosion
- Particularly good for vision, speech, RL and NLP

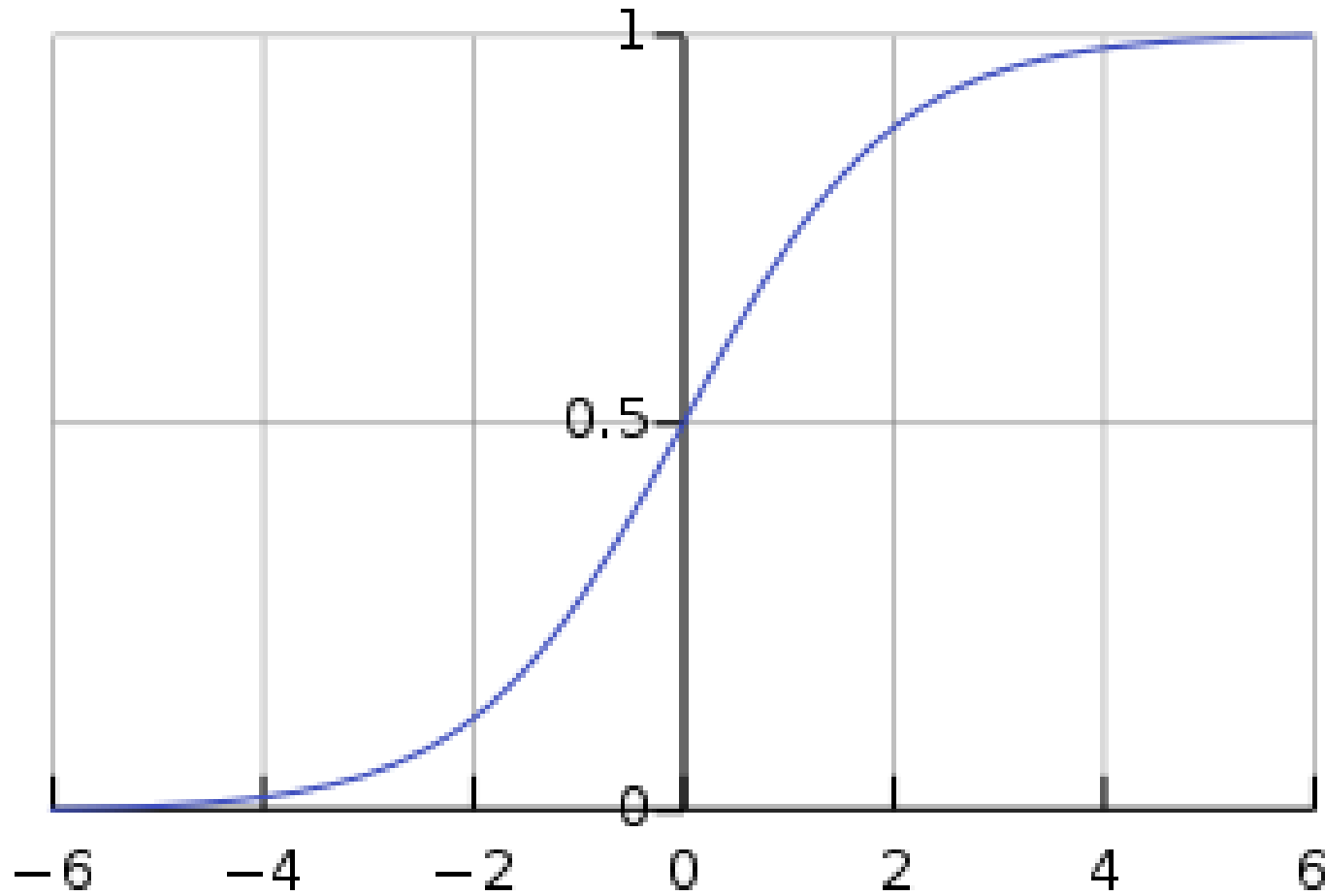


WHAT ARE NEURAL NETWORKS?



WHAT ABOUT "DEEP" NEURAL NETWORKS?

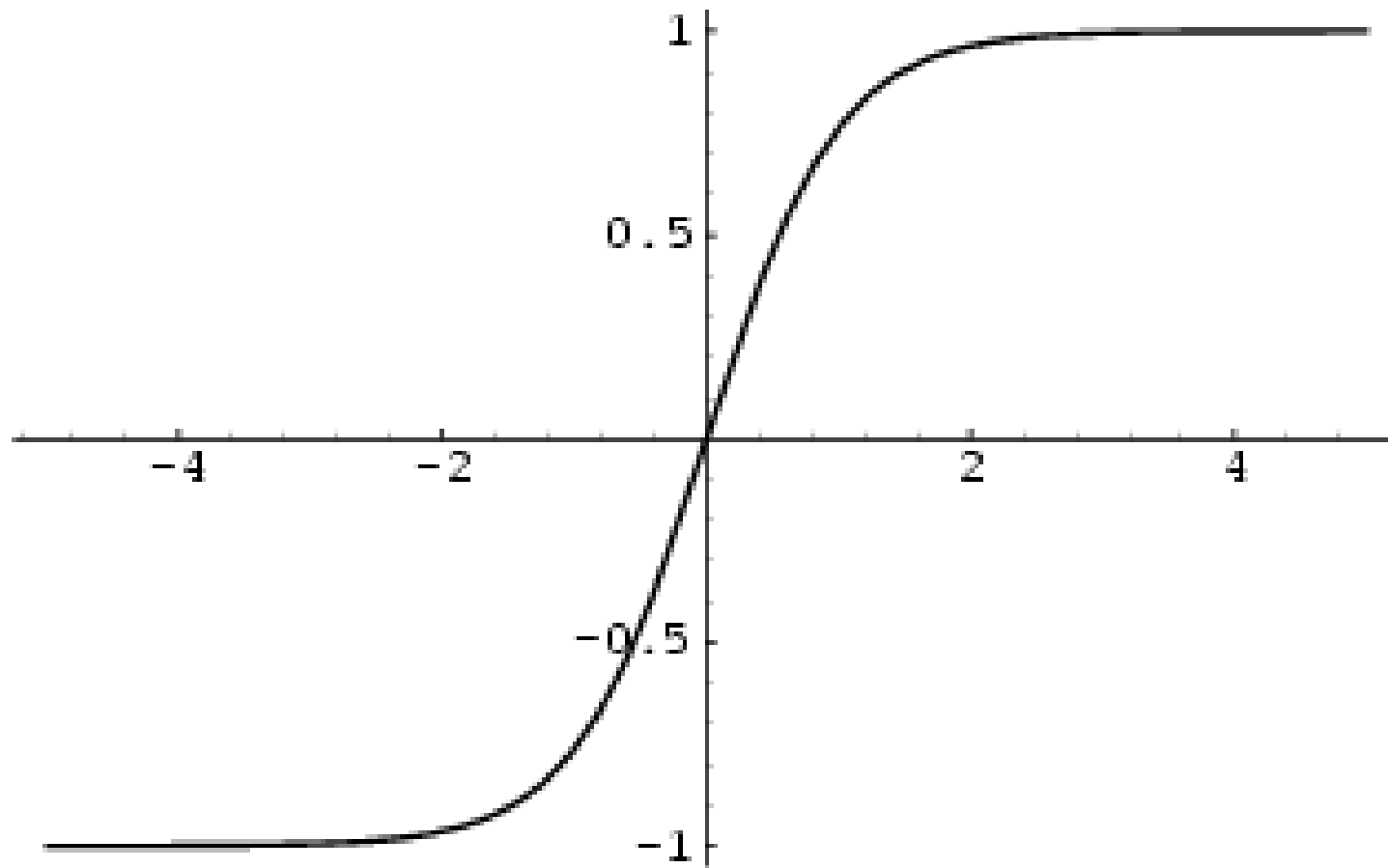




$$\frac{e^x}{e^x + 1}$$

SIGMOID SQUASHING FUNCTION



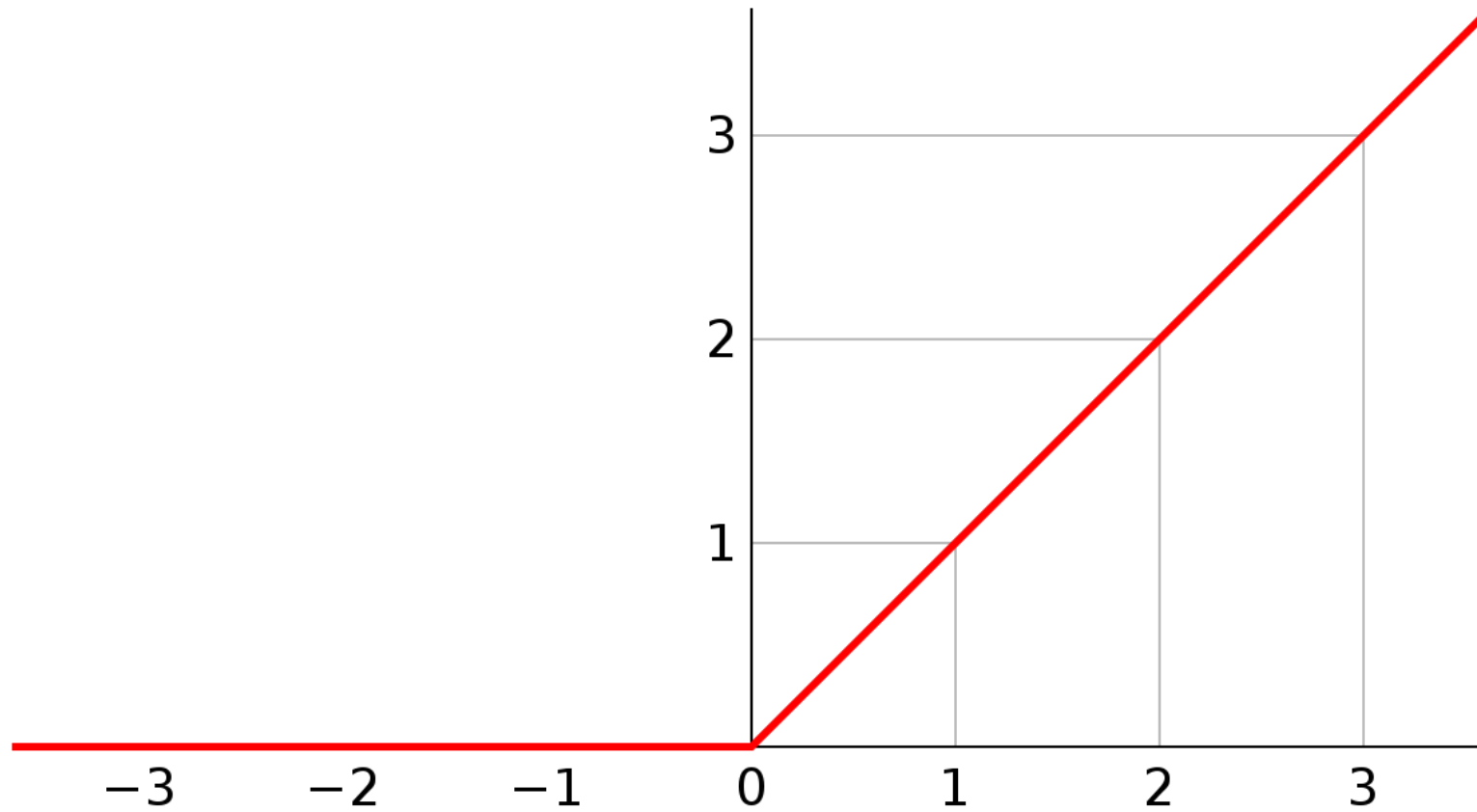


$$\frac{1 - e^{-2x}}{1 + e^{-2x}}$$

TANH SQUASHING FUNCTION



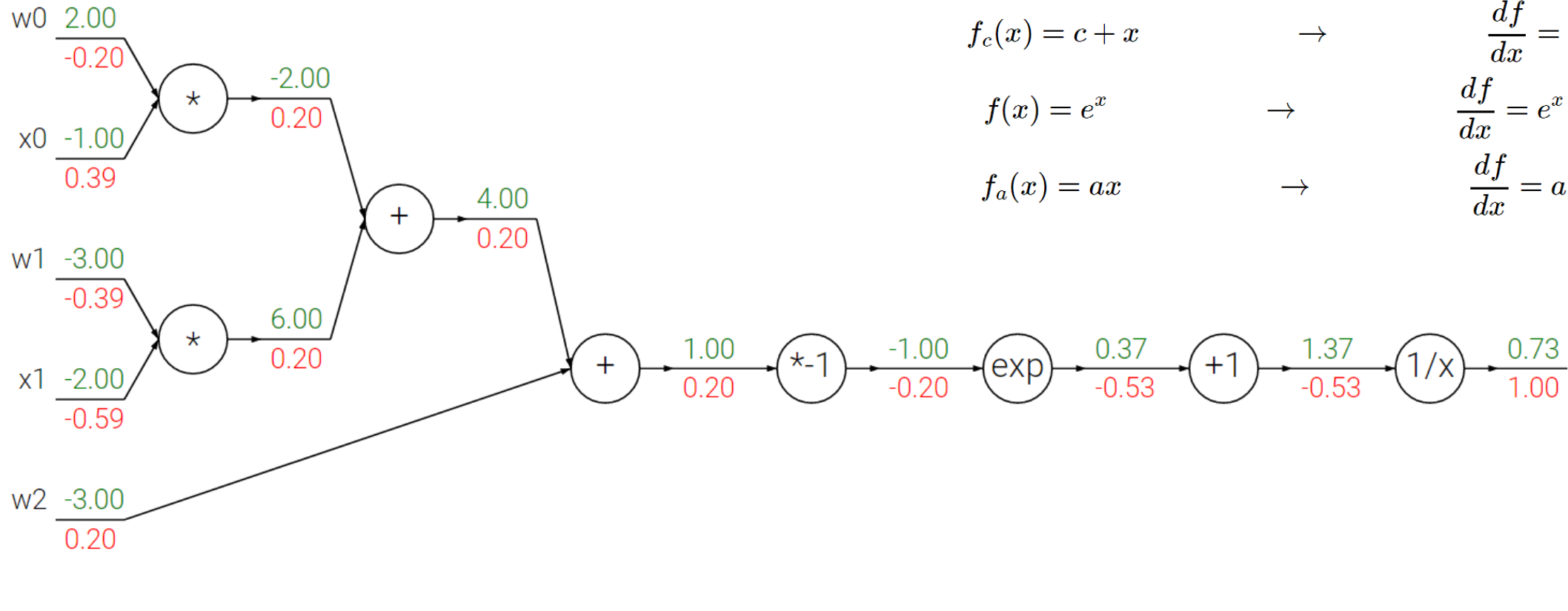
$$f(x) = x^+ = \max(0, x)$$



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.

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

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

learning rate error e_k derivative of output activation ϕ_k'

signal δ_k

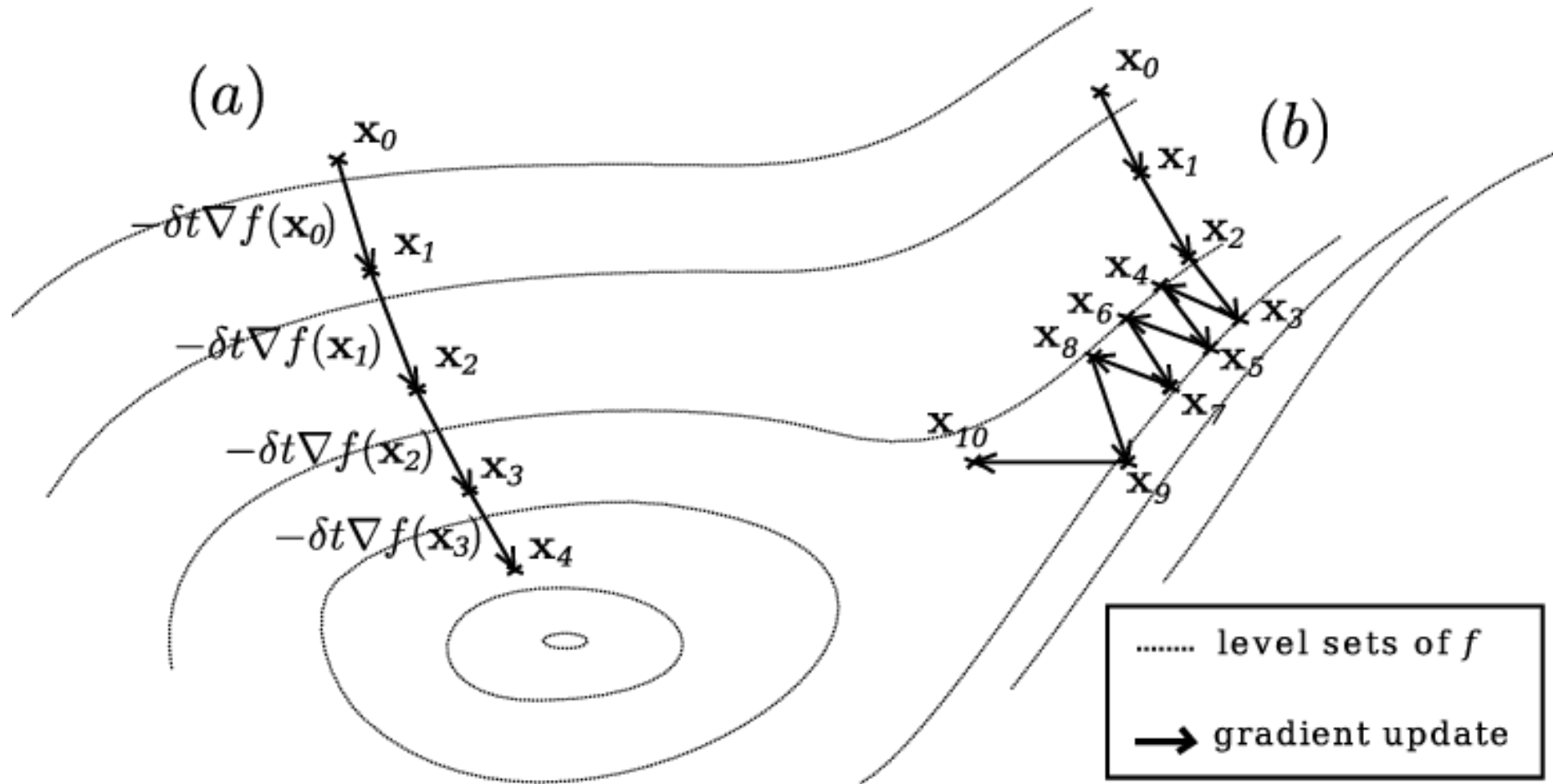
WEIGHT UPDATE


```
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



WHAT IS CNTK?

DECLARITIVELY DESCRIBE AND TRAIN DEEP
NEURAL NETWORKS

DOES ALL THE HARD WORK FOR YOU

80% INTERNAL MS DL WORKLOADS USE
CNTK

1ST CLASS ON LINUX, WINDOWS, DOCKER

C#, PYTHON, COMMANDLINE

KERAS BINDINGS



<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)

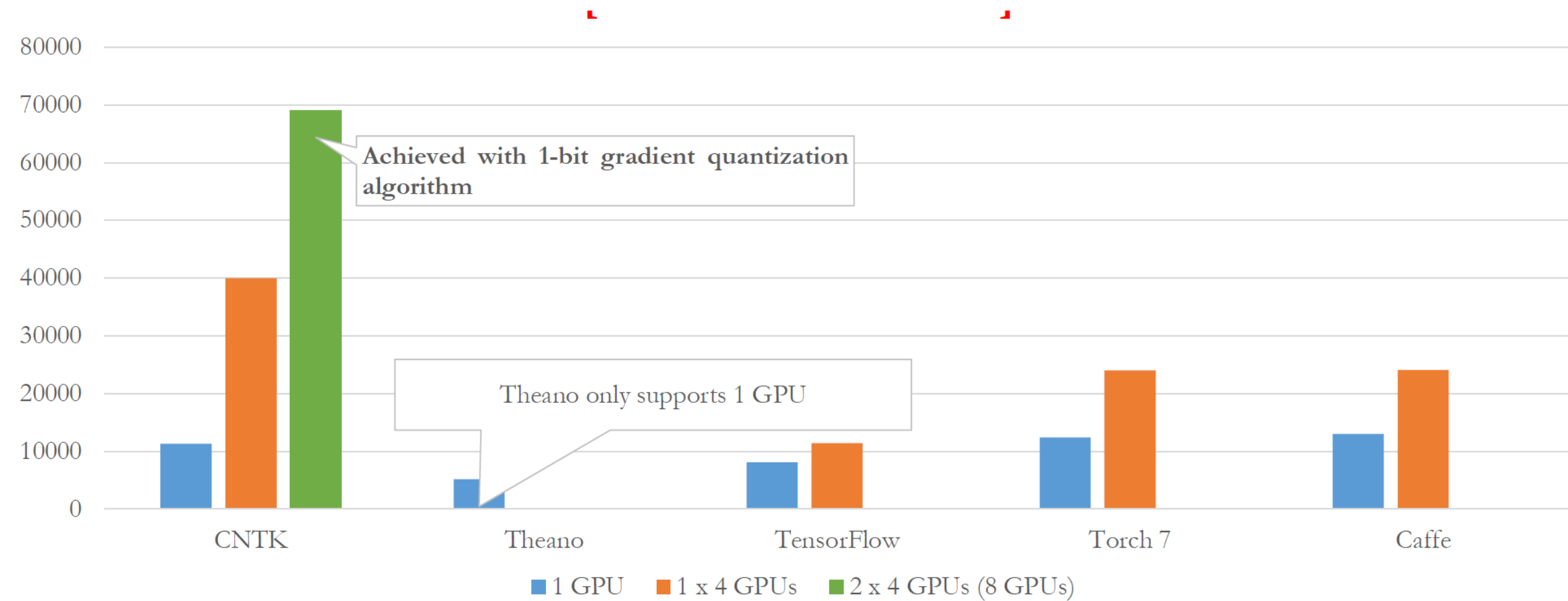
TensorFlow: 1.0(4ac9c09)

Torch: 7(748f5e3)

	Caffe	CNTK	MxNet	TensorFlow	Torch
FCN5 (1024)	55.329ms	51.038ms	60.448ms	62.044ms	52.154ms
AlexNet (256)	36.815ms	27.215ms	28.994ms	103.960ms	37.462ms
ResNet (32)	143.987ms	81.470ms	84.545ms	181.404ms	90.935ms
LSTM (256) (v7 benchmark)	-	43.581ms (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"



DEEP LEARNING ON AZURE CLOUD

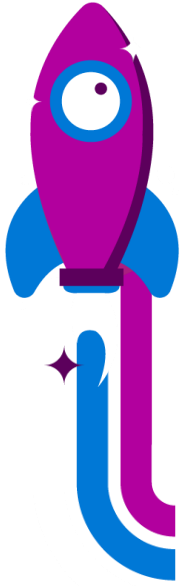
- Data Science Virtual Machine (Ubuntu and Windows)
- Batch AI Training Service
- AzureML supports some deep learning workloads
- R Server supports some deep learning



DEMO

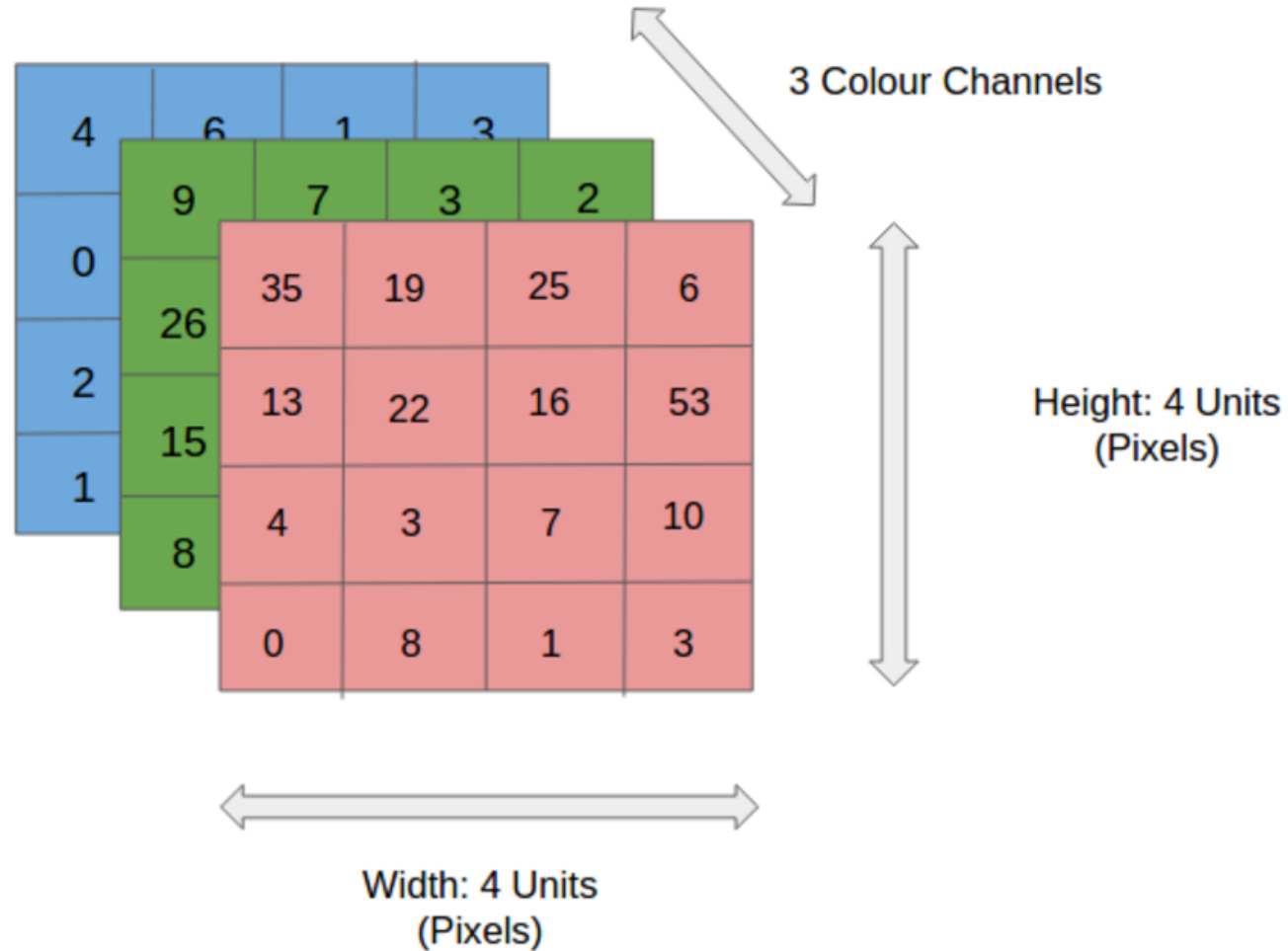
CNTK IRIS DEMO





WHAT ABOUT VISION AND NATURAL LANGUAGE PROCESSING?

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

$$(50 \times 30) + (50 \times 30) + (50 \times 30) + (20 \times 30) + (50 \times 30) = 6600$$



Visualization of the receptive field

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	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 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



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



8	12
46	100

(i)

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



35	63
9	10

(iii)

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



37	14
29	54

(ii)

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



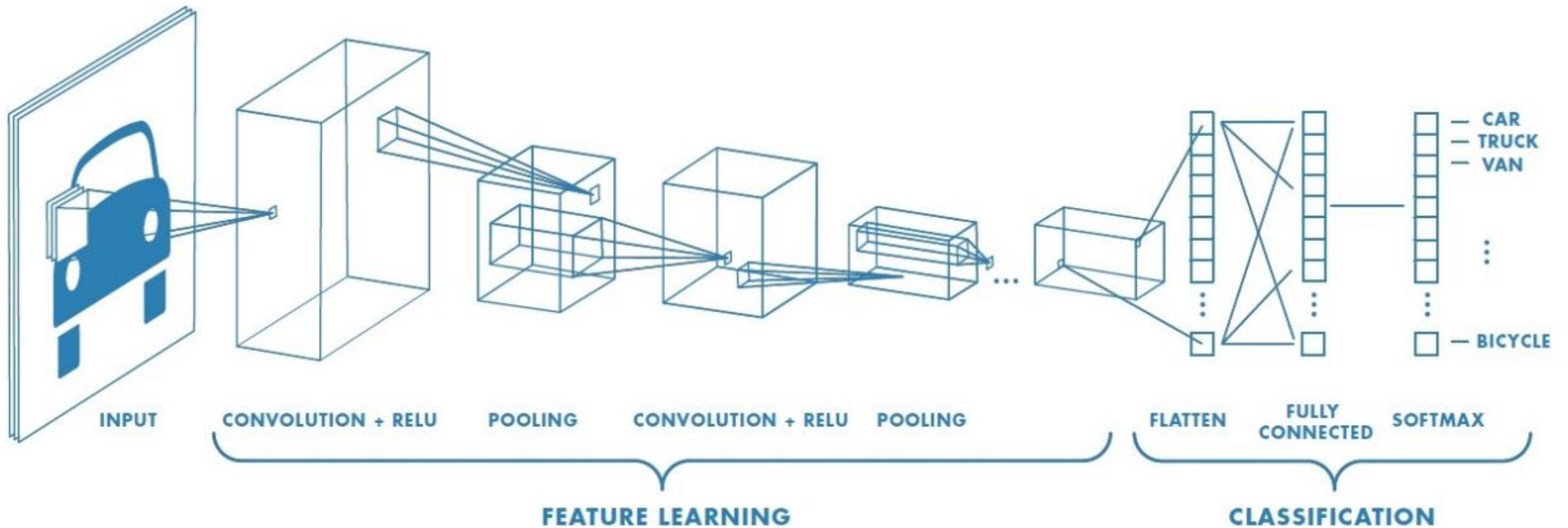
35	25	63
22	22	63
9	8	10

(iv)

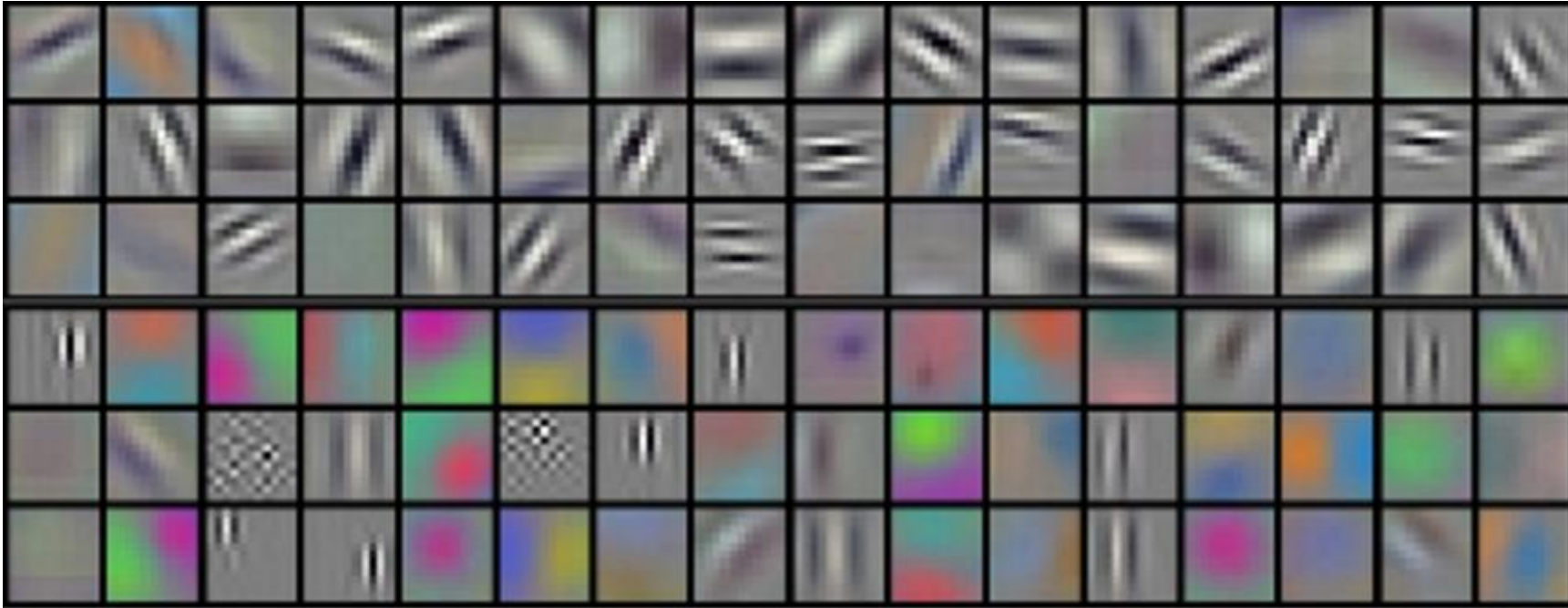
POOLING



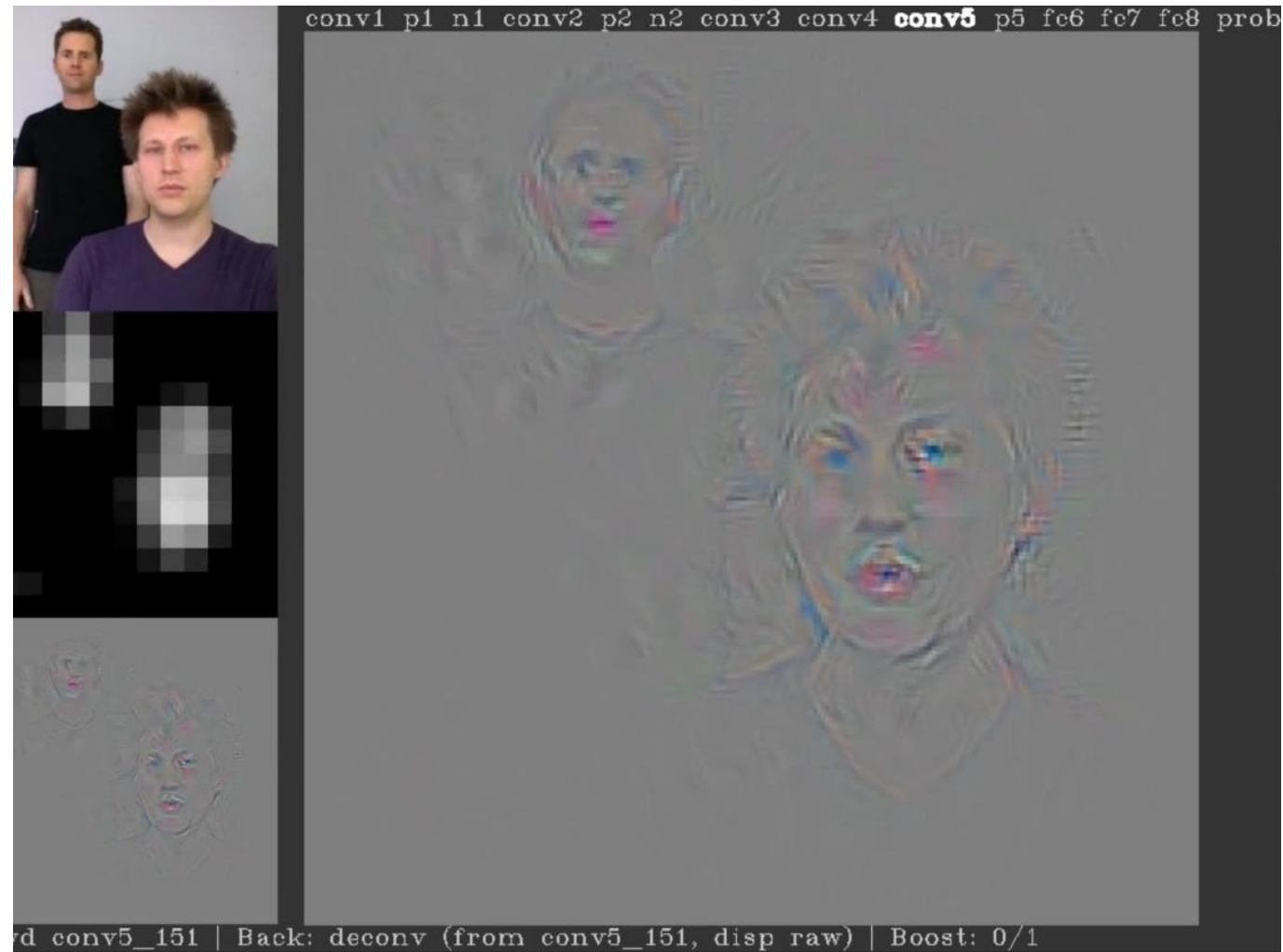
Image Classification Example



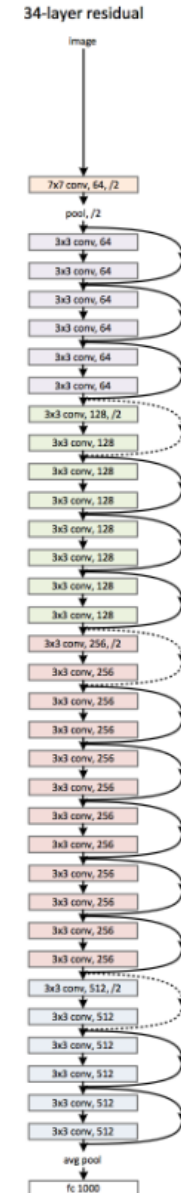
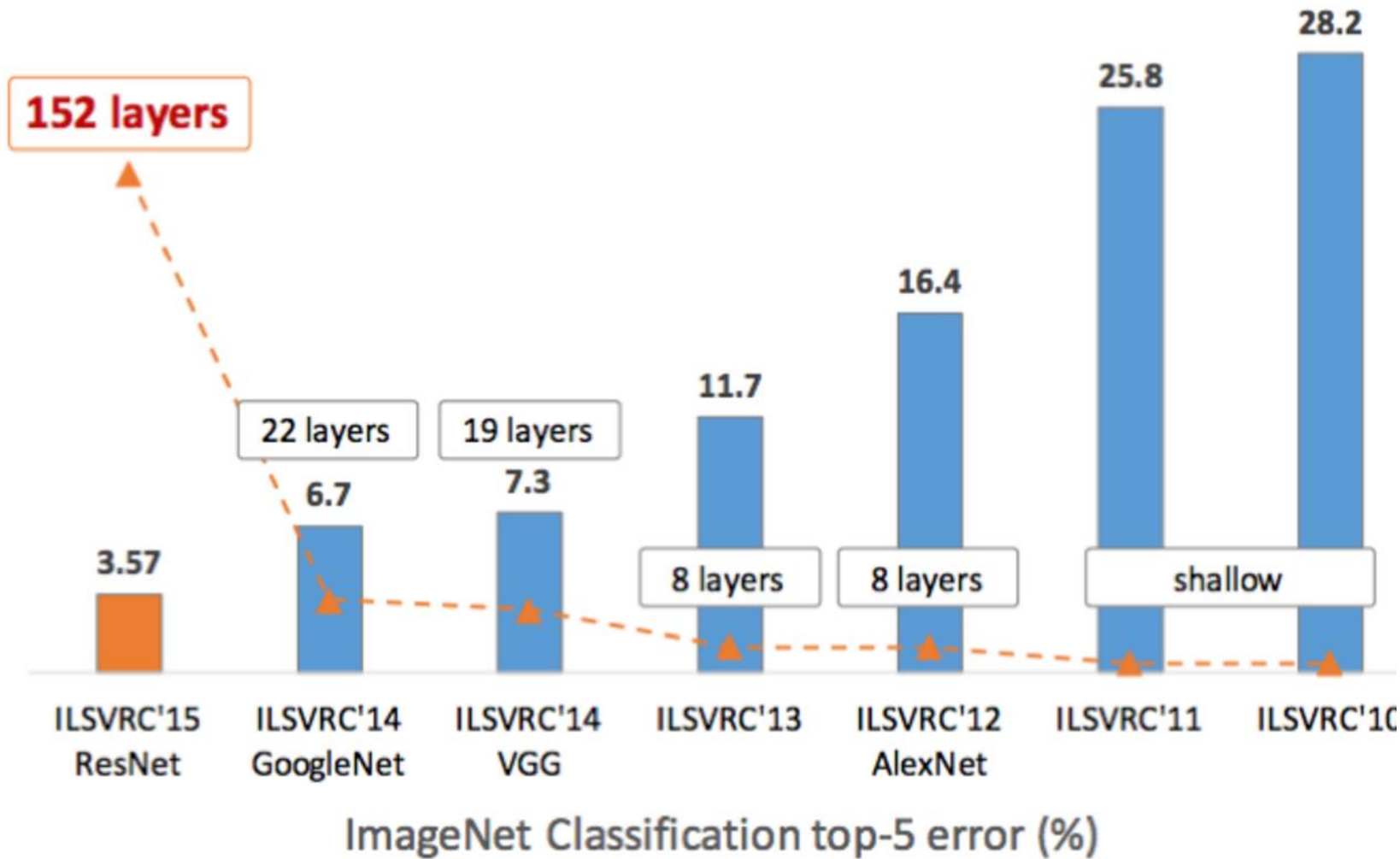
VISUALISING THE FILTERS



DEEP VISUALISATION TOOLBOX



RESOLUTION OF DEPTH



DEMO

CNTK MNIST DEMO



THANK YOU

tim.scarfe@microsoft.com

