ST449 Artificial Intelligence and Deep Learning

Lecture 1

Course Overview

Milan Vojnovic

https://github.com/lse-st449/lectures

Topics of this lecture

- Course organization and resources
- Overview of lecture/seminar topics
- Overview of applications and resources
- Information for Seminar class 1

When

- Lectures:
  - Lent Term, Wednesday 11:00-13:00
  - NAB 1.04

- Seminars:
  - Lent Term, Friday 14:30-16:00
  - 32L.LG.18

Your team

- Lectures:

Milan Vojnovic, Department of Statistics
COL 5.05
m.vojnovic@lse.ac.uk
Office hours: by appointment

- Seminars:

Tianlin Xu, Department of Statistics
COL 5.03
t.xu12@lse.ac.uk
Office hours: Thursday 11:00-12:00

Resources

- Course handout: https://lse-st449.github.io
  - Contains a short summary of topics covered in each week of the course

- GitHub lse-st449 organization: https://github.com/lse-st449
  - Used for access to lecture and seminar material
  - Homework assignments (GitHub classroom)
  - Project repositories

- Some useful links:
  - Lectures: https://github.com/lse-st449/lectures
  - Notifications: https://github.com/lse-st449/lectures/blob/master/README.md
  - Projects: https://github.com/lse-st449/lectures/blob/master/Projects.md

What: course outline

Part I – neural networks

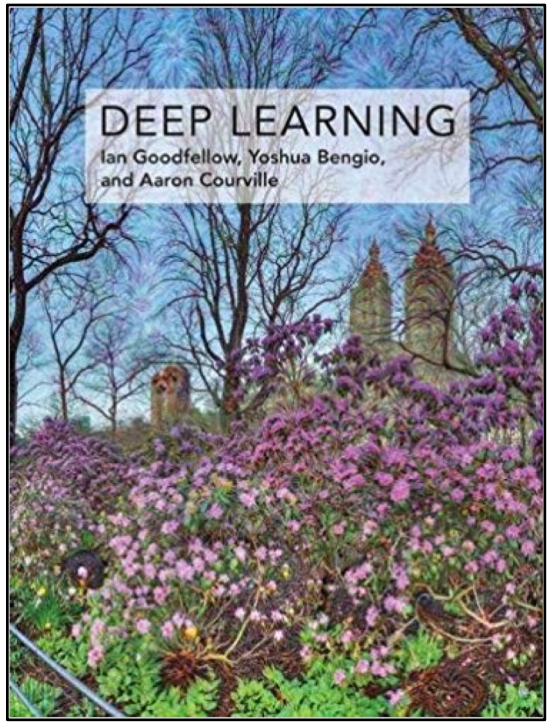
1. Course overview
2. Introduction to neural networks
3. Training neural networks
4. Convolutional neural networks
5. Sequence modeling

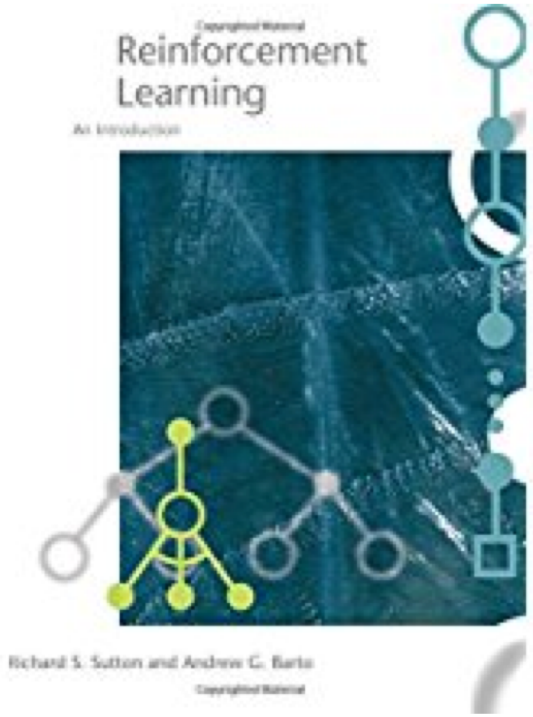
Part II – reinforcement learning

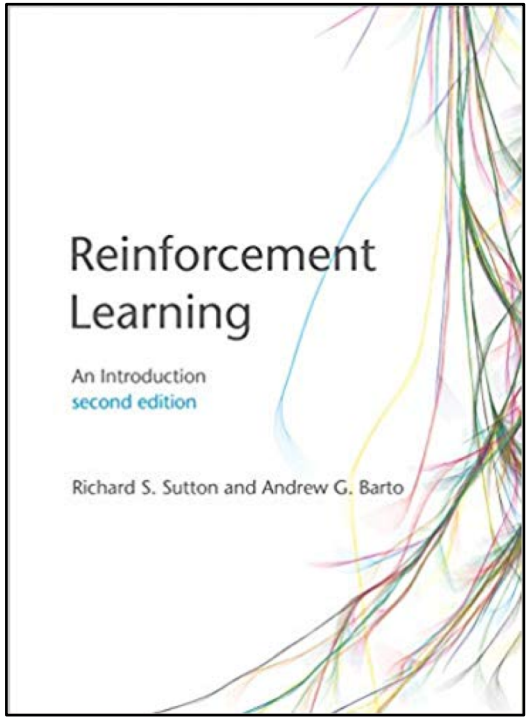
6. Introduction to reinforcement learning
7. Dynamic programming and Monte Carlo methods
8. Temporal difference methods and eligibility traces
9. Generalization and function approximation
10. Policy gradient

Resources

- Selected chapters from books



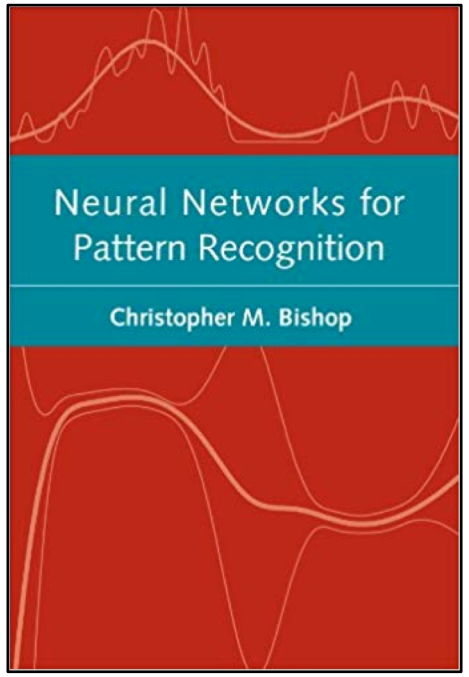


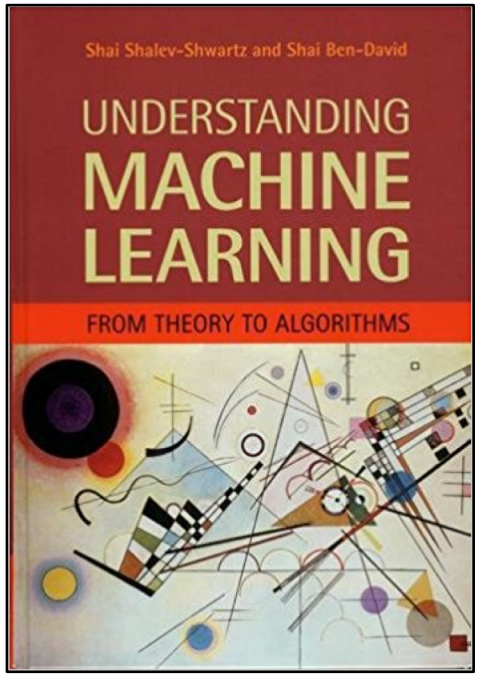


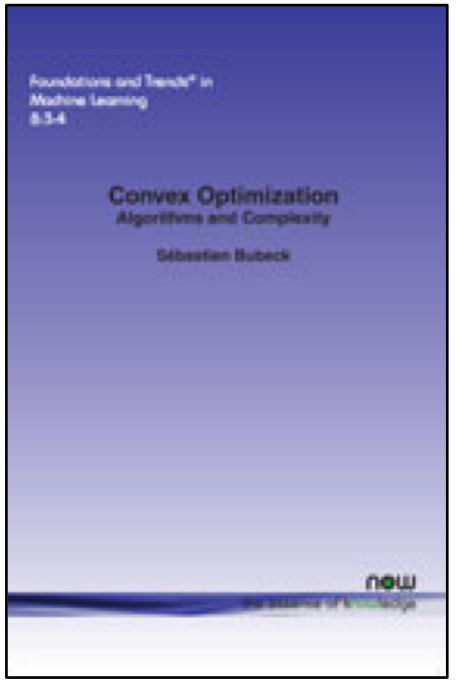
https://www.deeplearningbook.org

http://incompleteideas.net/book/the-book-2nd.html

Additional resources







http://sbubeck.com/book.html

Additional resources

- Research papers !
  - Ex published at machine learning conferences (NeurIPS, ICML, ICLR, …)
  - Active research area
  - Many neural network architectures have been developed recently

Software used in this course

- Python
- Jupyter notebooks
- TensorFlow: https://www.tensorflow.org
  - An open source machine learning framework for dataflow graph computing
- OpenAI Gym: https://gym.openai.com
  - A toolkit for developing and comparing reinforcement learning algorithms
- Google services
  - Colaboratory
  - Google Cloud Platform (sponsored coupon USD 50.00 per student)

Evaluation and marking

- Continual assessment
  - Two assignments each 10% of the mark
  - First assignment in lecture #4
  - Second assignment in lecture #9

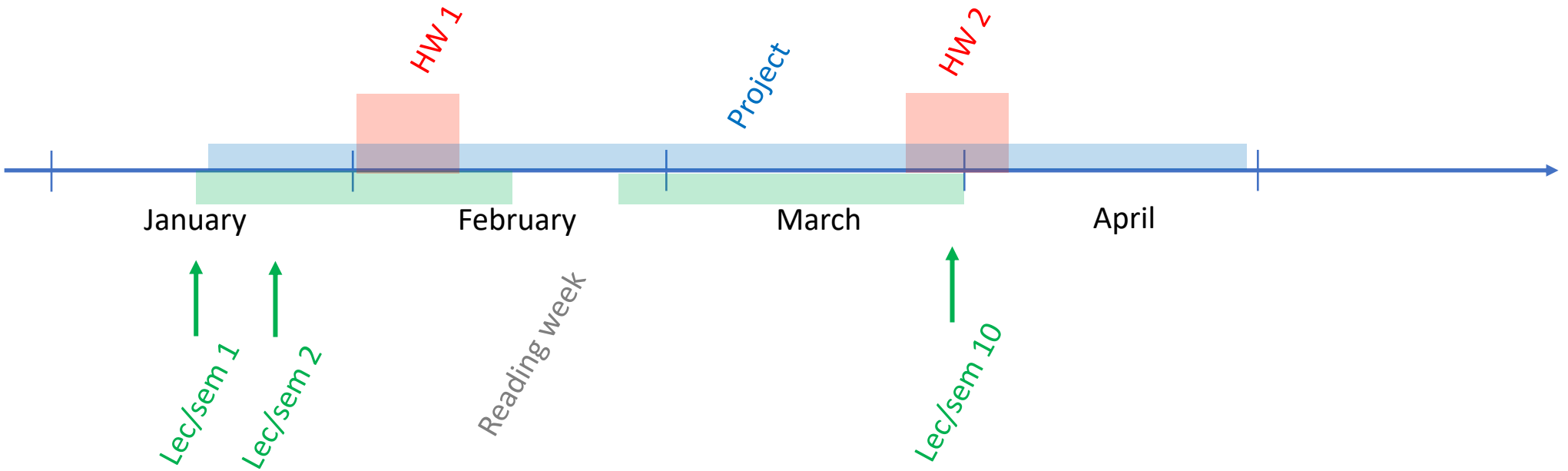
- Course project
  - 80% of the mark
  - Project topic discussions with the lecturer

- Important dates:
  - **Feb 27** first assignment solution
  - **Apr 10** second assignment solution
  - **Apr 30** project report

Course projects

- Individual project on a topic falling in the scope of the course
  - Focus on methodological aspects, principles, and implementation
- Allowing for
  - Accounting for individual preferences
  - Allowing to explore a given topic in more depth
  - Allowing to gain hands-on experience in coding a solution
- Some hints for project topics:
  - Implement and evaluate a neural network for a classification task
  - Implement and evaluate a neural network for a sequence modelling task
  - Implement and evaluate a solution for a reinforcement learning task
  - More hints in the GitHub lse-st449 file Projects.md

Course workflow



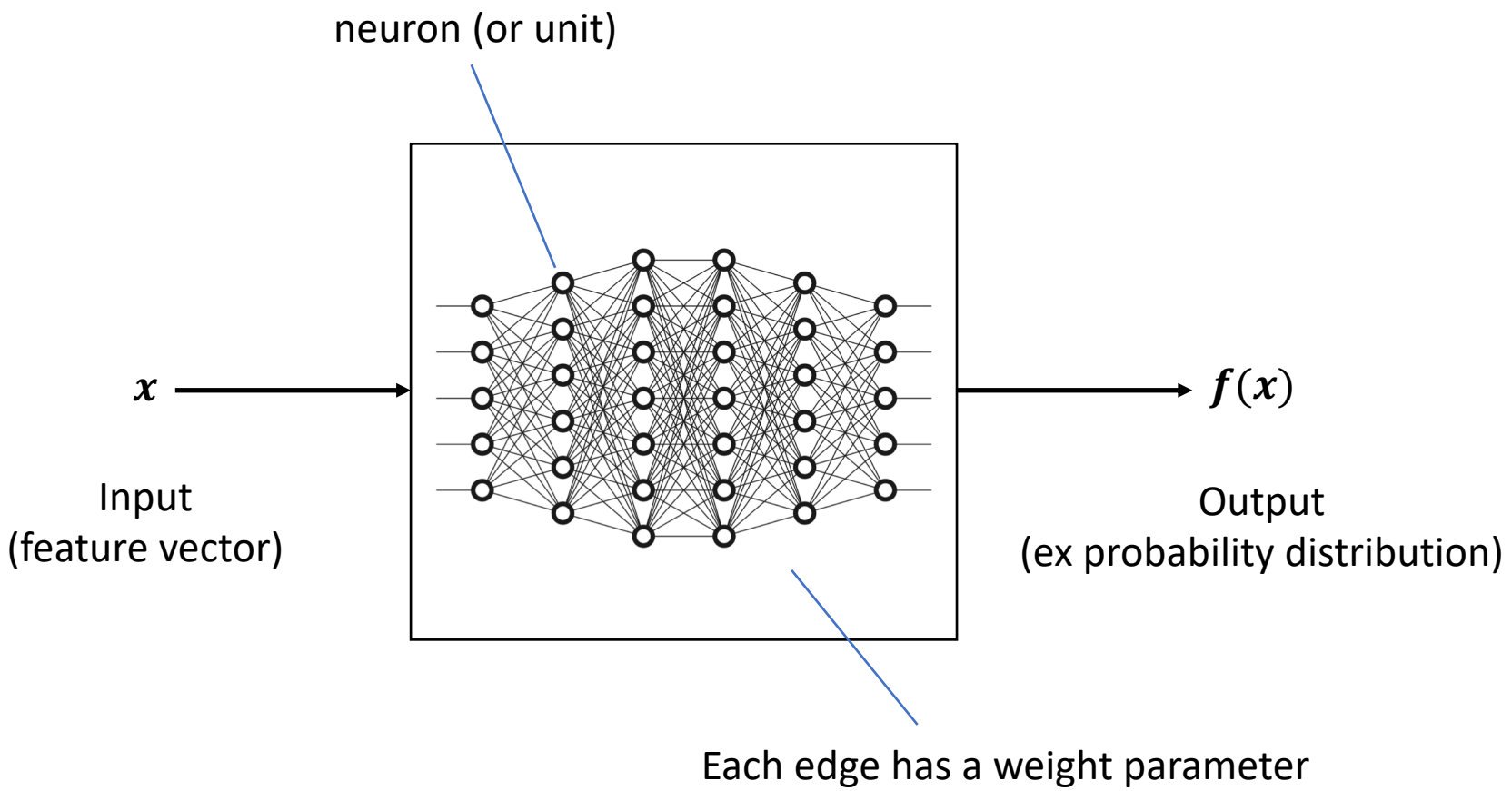
Course overview

- Neural networks
  - Basic principles
  - Optimization for training neural networks
  - Convolutional neural networks
  - Sequence models

- Reinforcement learning
  - Basic principles
  - Tabular solution methods
  - Generalization and function approximation

- Check the handout: https://lse-st449.github.io

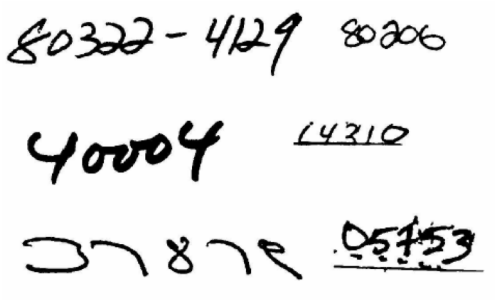
One-slide introduction to neural networks

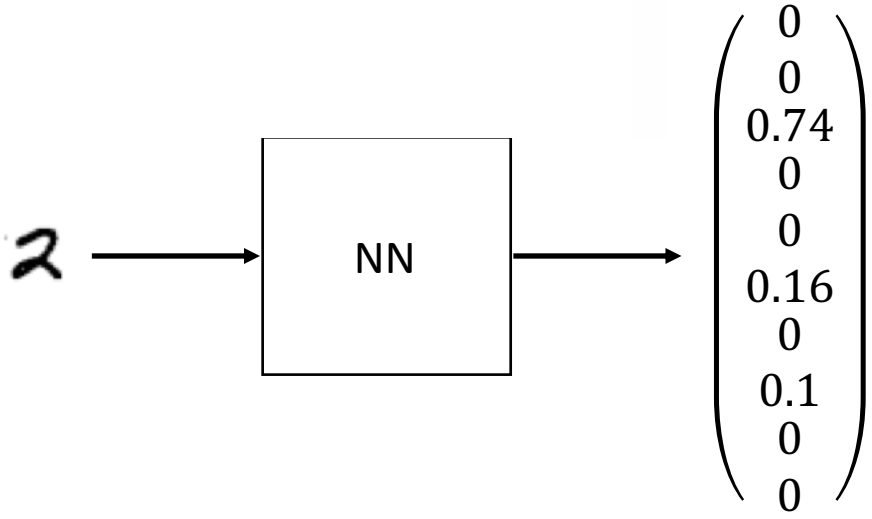


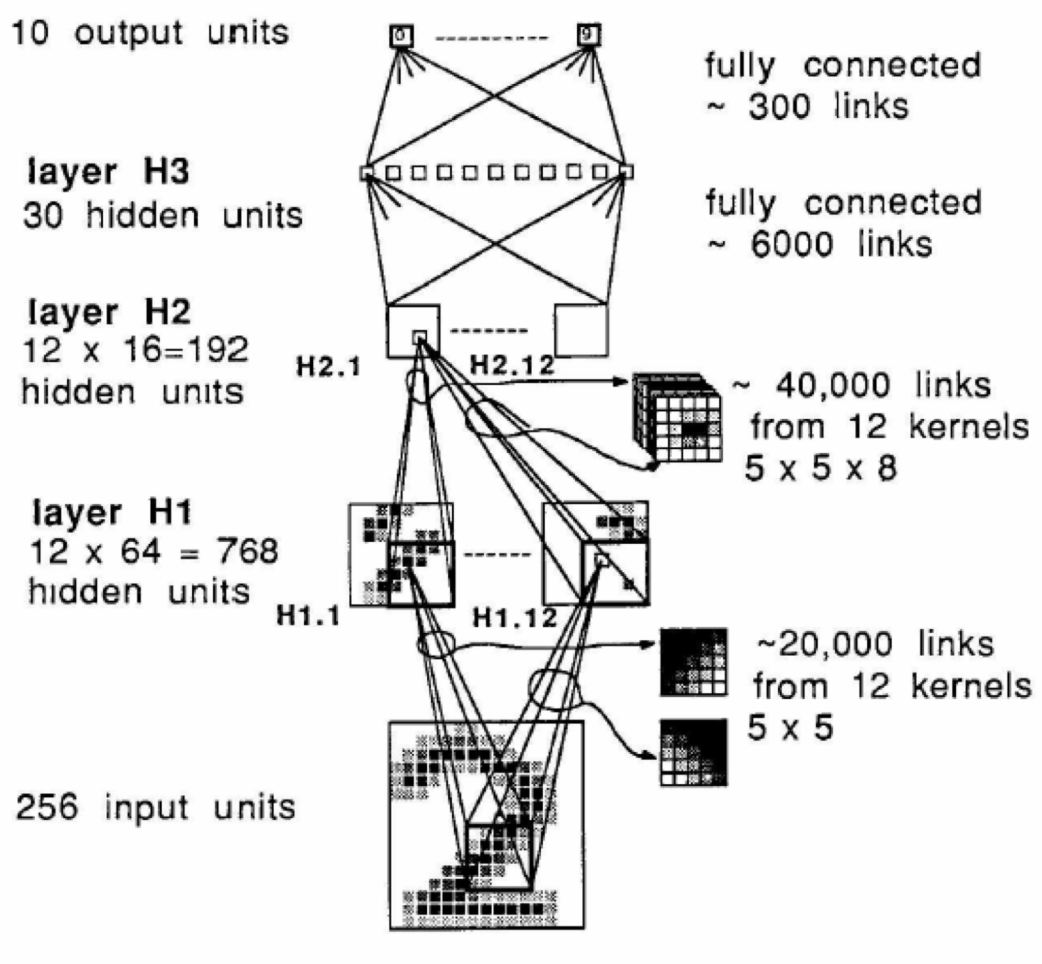
- Supervised learning: fitting network parameters using a set of training examples

Handwritten zip code recognition

- LeCun 1980s



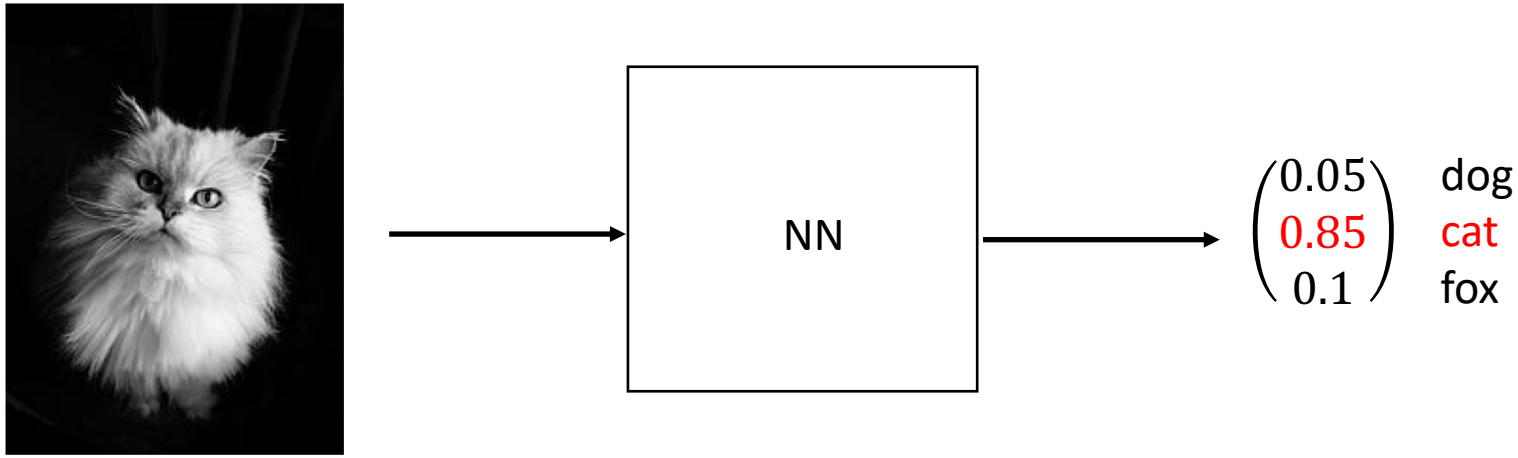




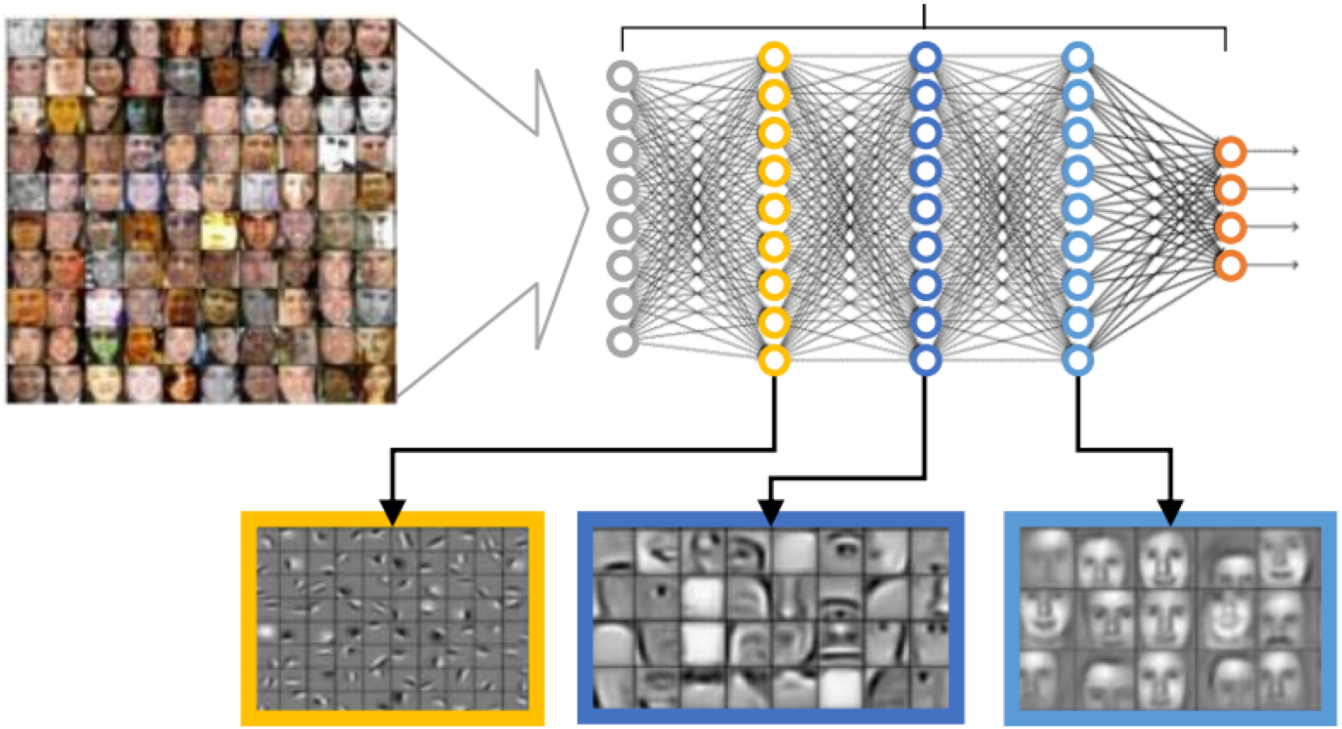
MNIST database of handwritten digits

- http://yann.lecun.com/exdb/mnist/

Image classification

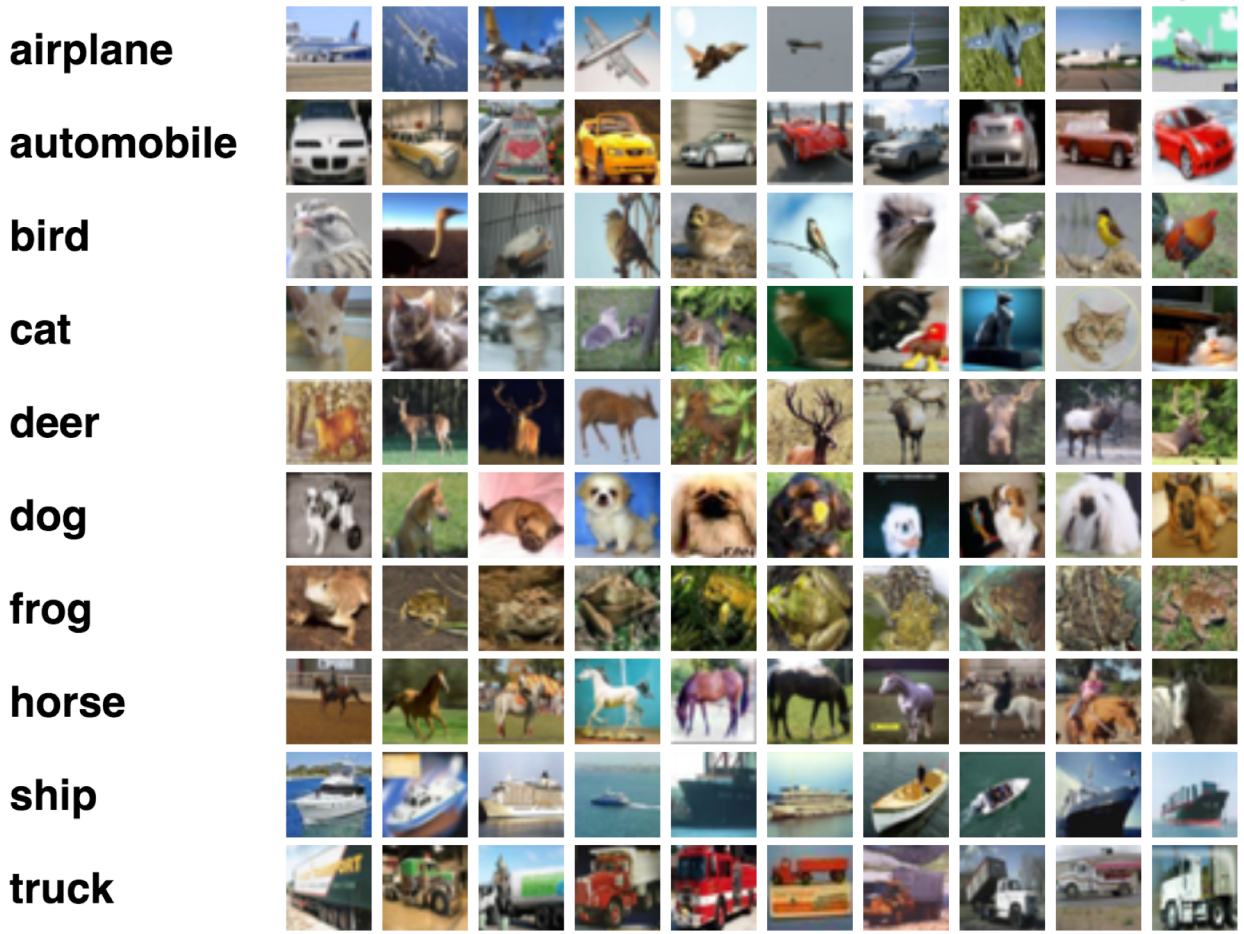


Representation learning



- Feedforward neural network example:
  - The last layer is typically a linear classifier
  - Other layers learn a representation for classification

CIFAR



- https://www.cs.toronto.edu/~kriz/cifar.html
- CIFAR-10: 60000 32x32 colour images in 10 classes, 6000 images per class
- CIFAR-100: 100 classes, 600 images each; grouped into 20 super-classes

IMAGENET: Large Scale Visual Recognition Challenge

- ILSVRC: http://www.image-net.org/challenges/LSVRC/

- Evaluation of algorithms at large scale for
  - Image classification
  - Object detection

- Motivation:
  - Allow researchers to compare progress in detection across a variety of objects
  - Measure the progress of computer vision for large scale image indexing for retrieval and annotation

- Run since 2010

ILSVRC tasks

- Image level annotation (image classification): a binary label for the presence or absence of an object class in the image
  - Ex ”there are cars in this image” but “there are no tigers”

Image classification

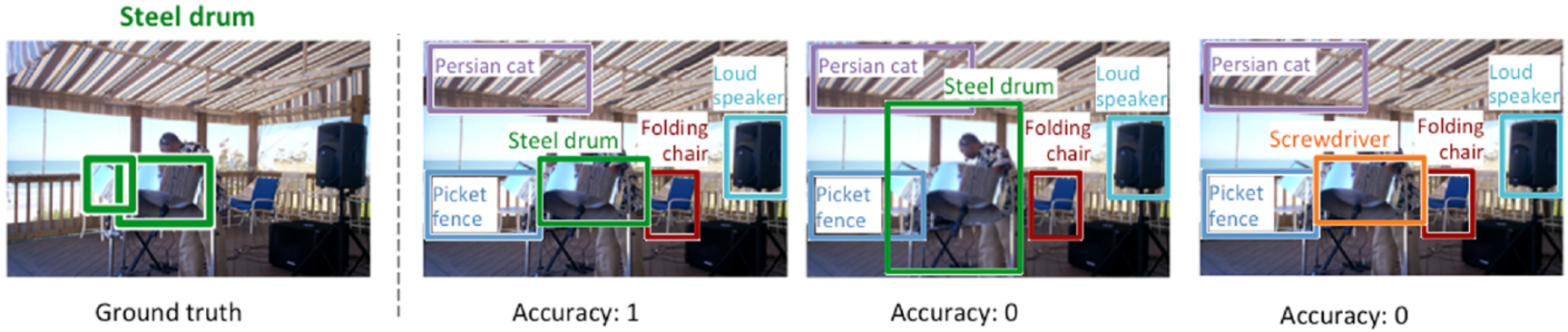
Ground truth

Steel drum Folding chair Loudspeaker	Scale T-shirt Steel drum Drumstick Mud turtle	Scale T-shirt Giant panda Drumstick Mud turtle
Accuracy: 1	Accuracy: 1	Accuracy: 0

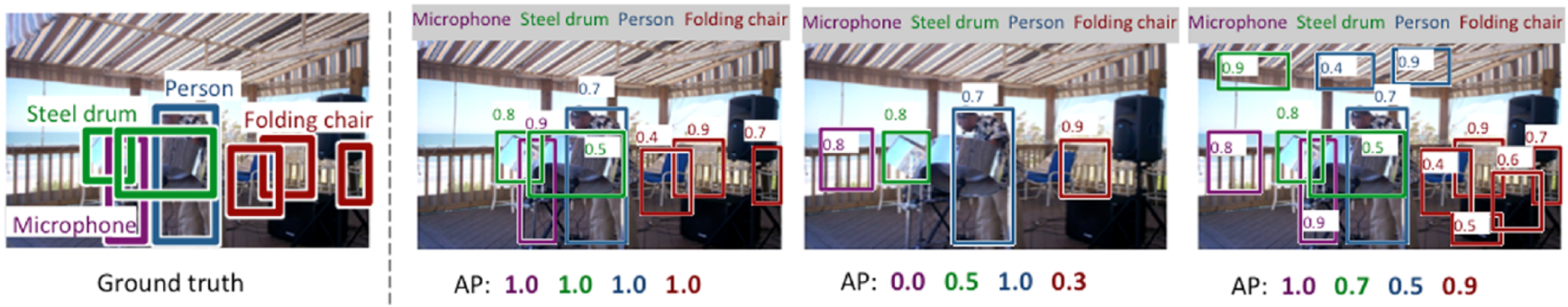
ILSVRC tasks

- Object level annotation (object localization): annotation of a tight bounding box and class label around an object instance in the image
  - Ex ”there is a screwdriver centered at position (20,25) with width 50 pixels and height 30 pixels”

Single-object localization:



Object detection:



Persian cat

A long-haired breed of cat

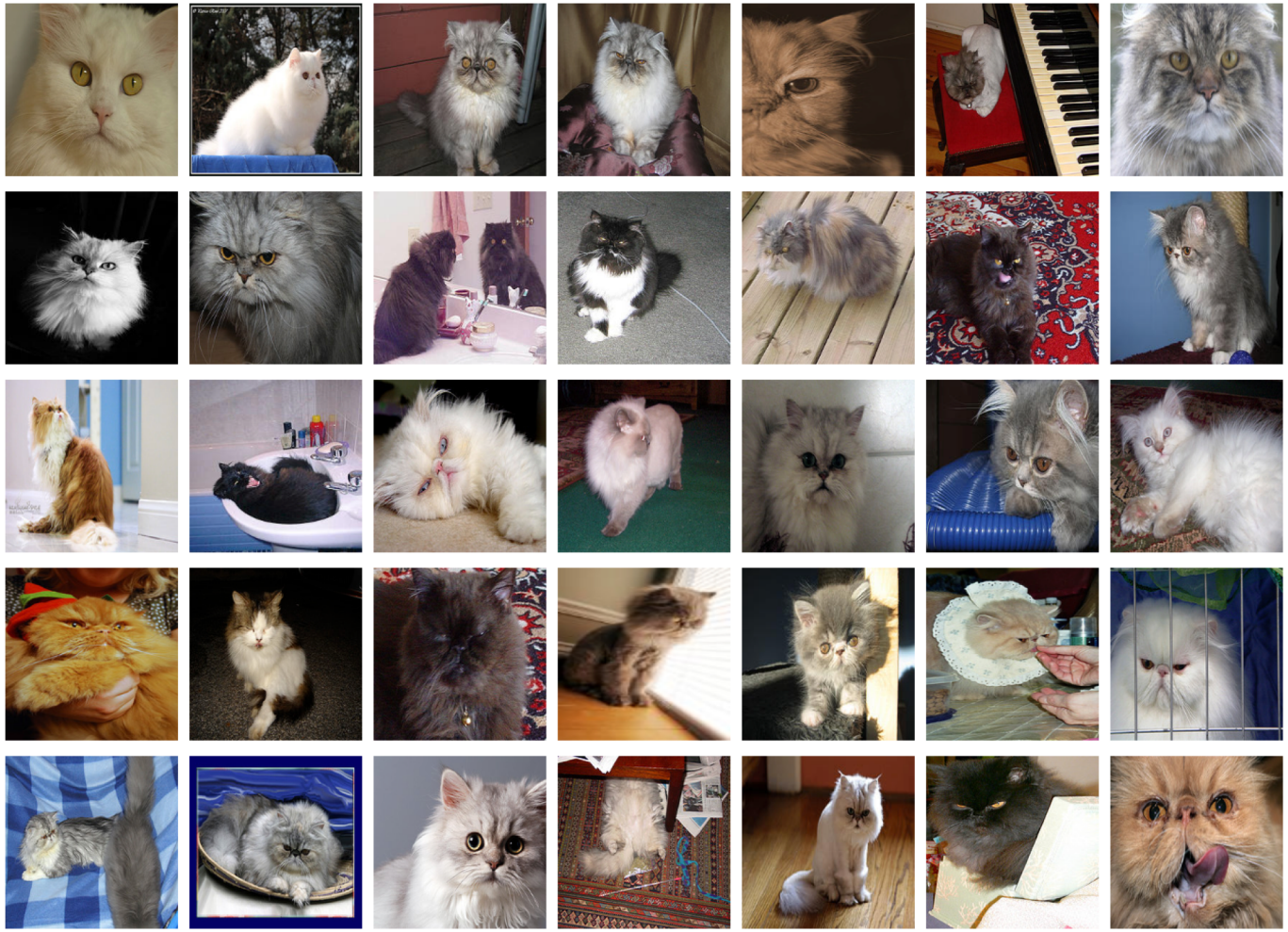
1662 pictures | 59.56% Popularity Percentile | Wordnet IDs

http://image-net.org

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
  - plant, flora, plant life (4486)
  - geological formation, formation (1
  - natural object (1112)
  - sport, athletics (176)
  - artifact, artefact (10504)
  - fungus (308)
  - person, individual, someone, someb
  - animal, animate being, beast, brute
    - invertebrate (766)
    - homeotherm, homoiotherm, hor
    - work animal (4)
    - darter (0)
    - survivor (0)
    - range animal (0)
    - creepy-crawly (0)
    - domestic animal, domesticated
      - domestic cat, house cat, Feli
        - Egyptian cat (0)
        - Persian cat (0)
        - kitty, kitty-cat, puss, puss
        - tiger cat (0)
        - Angora, Angora cat (0)
        - tom, tomcat (1)
        - Siamese cat, Siamese (1)
        - Manx, Manx cat (0)
        - Maltese, Maltese cat (0)
        - tabby, queen (0)
        - Burmese cat (0)

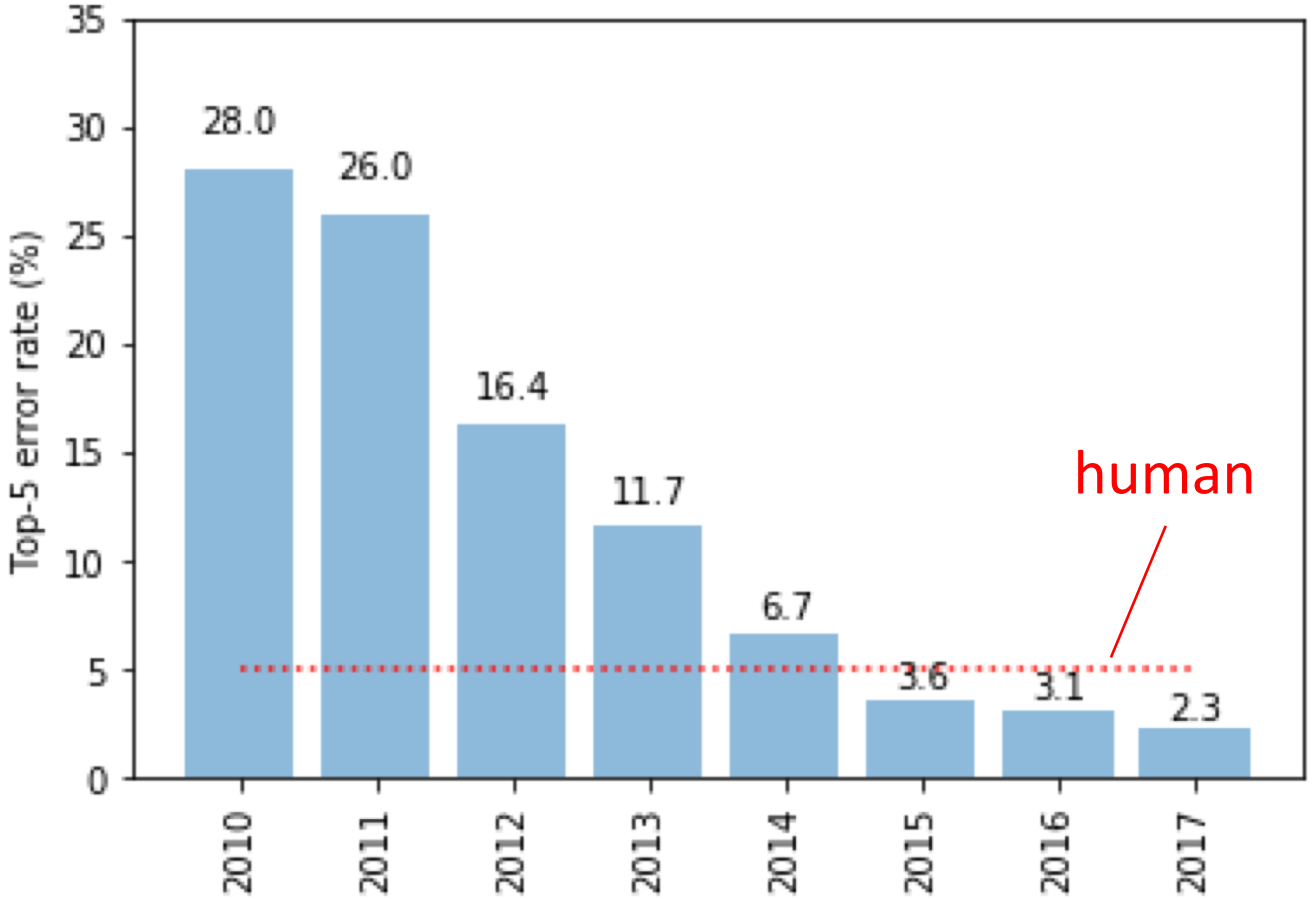
Treemap Visualization | Images of the Synset | Downloads



*Images of children synsets are not included. All images shown are thumbnails. Images may be subject to copyright.

Prev 1 2 3 4 5 6 7 8 9 10 ... 71 72 Next

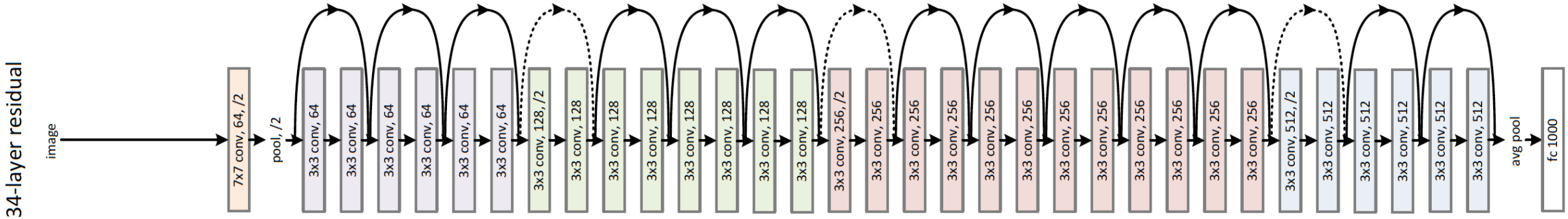
Large-scale vision recognition challenge error rates



2010
2011 XRCE
2012 AlexNet
2013 ZFNet
2014 VGG, GoogLeNet
2015 ResNet
2016 GoogLeNet-v4
2017

Example neural network architecture

- ResNet, LSRVRC 2015
- 152 layers



Speech recognition

1784

Wolfgang von Kempelen creates the **Acoustic-Mechanical Speech Machine** in Vienna

Thomas Edison invents the first dictation machine

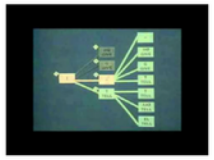
1879

1952

Bell Labs releases **Audrey**, capable of recognizing spoken digits with **90% accuracy** - but only when spoken by its inventor

IBM Shoebox can understand 16 English words

1962



1971

Harpy, created at Carnegie Mellon University, can comprehend **1,011 words** - and some phrases

IBM Tangora, using the Hidden Markov Model, predicts upcoming phonemes in speech

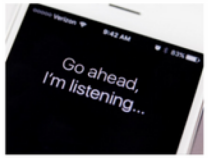
1986

2006

The National Security Agency (NSA) starts using speech recognition to isolate key words in recorded speech

Google launches a voice search app, bringing speech recognition to mobile devices

2008



2011

Apple announces Siri, ushering in the age of the voice-enabled digital assistant

ClickZ

Source: https://medium.com/swlh/the-past-present-and-future-of-speech-recognition-technology-cf13c179aaf

Speech recognition

- Modeling
  - Acoustic (ex convolutional neural networks architectures)
  - Language (ex recurrent neural networks with gated memory (LSTM) models)

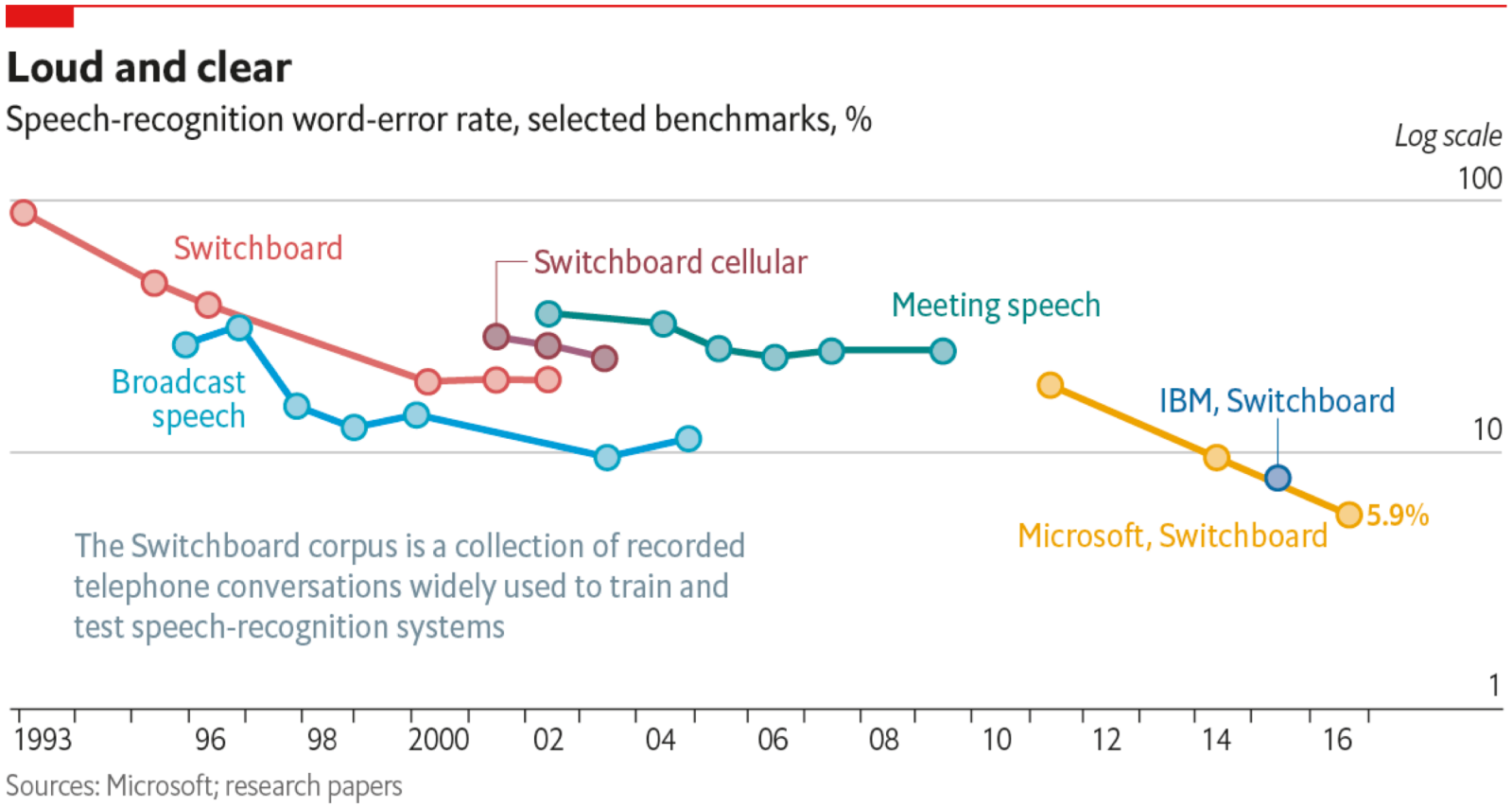
- Switchboard corpus word error rates
  - Mid 2000: 15%
  - Current: below 10%

- Improvements attributed to
  - Neural network representations of acoustic contexts in time and frequency
  - Learning representations of functional world similarity

2000 NIST Speaker Recognition Evaluation

- Part of an ongoing series of yearly evaluations conducted by NIST
  - Provide an important contribution to the direction of research efforts and the calibration of technical capabilities
  - Intended to be of interest to all researchers working on the general problem of text independent speaker recognition
  - https://catalog.ldc.upenn.edu/LDC2001S97
- Language: English
- Data: telephone speech
  - 10,328 single channel SPHERE files encoded in 8-bit mulaw containing a total of approximately 4.31 Gbytes of data covering 148.9 hours of conversational telephone speech collected by LDC

Speech recognition word error rates

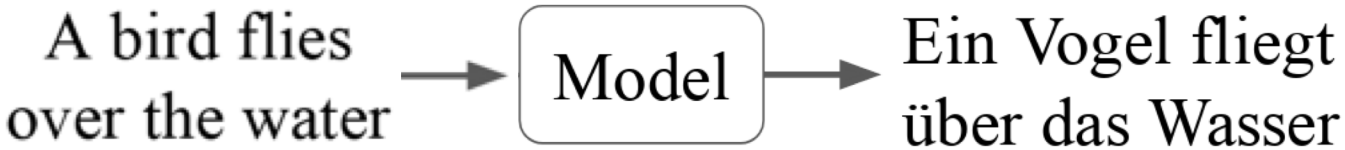


- Speech recognition: word-error rates, Finding a voice, The Economist: Technology Quarterly, 2017

Sequence modeling

- Machine translation
- Conversational dialogue
- …

Machine translation



- Translate an input text in language X to an output text in language Y

WMT tasks

- Translation
  - News: translation task on news text
  - Biomedical: evaluate systems on the translation of documents from the biomedical domain
  - Multimodal: generation of image descriptions in a target language
- Evaluation
  - Metrics: examine automatic evaluation metrics for machine translation
  - Quality estimation: examine automatic methods for estimating the quality of machine translation at run-time, without relying on reference translations
- Other
  - Automatic post-editing: automatic methods for correcting errors produced by an unknown machine translation (MT) system
  - Parallel corpus filtering: cleaning noisy parallel corpora

WMT: Conference on Machine Translation http://www.statmt.org/wmt18

Machine translation of news

- WMT 2018 example: http://www.statmt.org/wmt18/translation-task.html

German→English

	Ave. %	Ave. z	System
1	79.9	0.413	RWTH
	79.4	0.395	UCAM
	78.2	0.359	NTT
	77.3	0.346	ONLINE-B
	77.4	0.321	MLLP-UPV
	77.0	0.317	JHU
	76.9	0.315	UBIQUS-NMT
	76.7	0.310	ONLINE-Y
	75.7	0.268	ONLINE-A
	75.4	0.261	UEDIN
11	72.5	0.162	LMU-NMT
	72.2	0.149	NJUNMT-PRIVATE
13	65.2	−0.074	ONLINE-G
14	58.5	−0.296	ONLINE-F
15	45.4	−0.752	RWTH-UNSUPER
16	42.7	−0.835	LMU-UNSUP

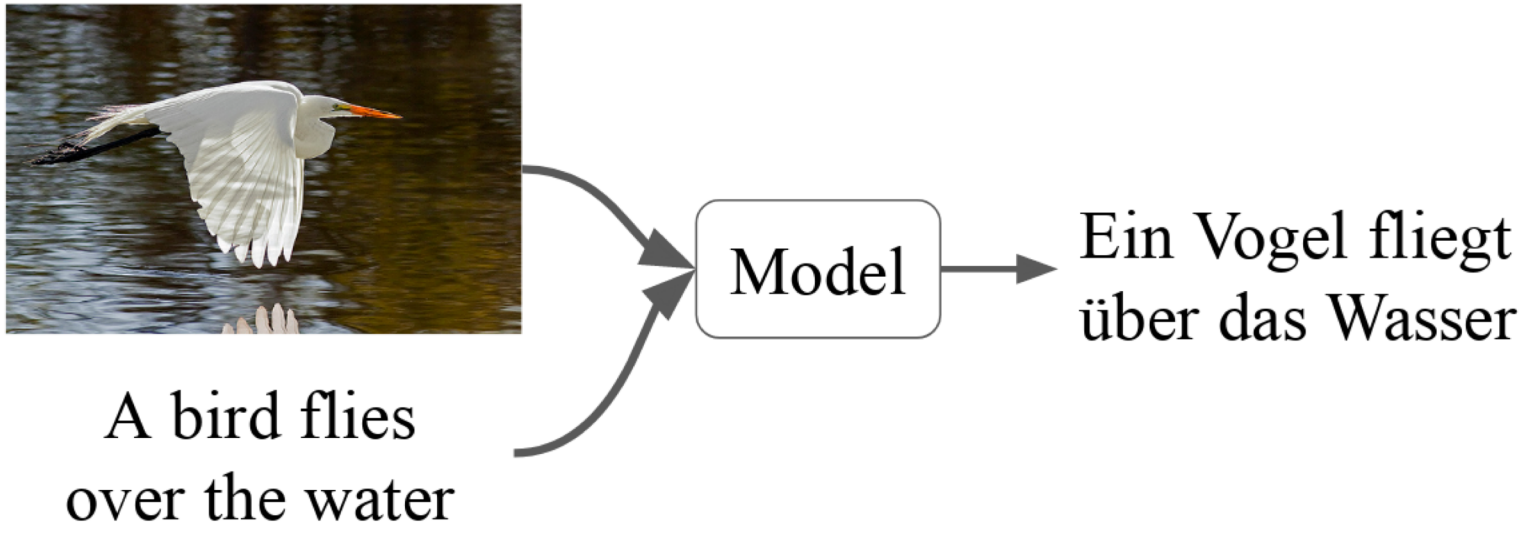
English→German

	Ave. %	Ave. z	System
1	85.5	0.653	FACEBOOK-FAIR ⋆
2	82.2	0.561	ONLINE-B
	81.9	0.551	MICROSOFT-MARIAN
	81.6	0.539	MMT-PRODUCTION
	82.3	0.537	UCAM
	80.2	0.491	NTT
	79.3	0.454	KIT
8	77.7	0.396	ONLINE-Y
	76.7	0.377	JHU
	76.3	0.352	UEDIN
11	71.8	0.213	LMU-NMT
12	67.4	0.060	ONLINE-A
13	53.2	−0.385	ONLINE-F
	53.8	−0.416	ONLINE-G
15	36.7	−0.966	RWTH-UNSUPER
16	32.6	−1.122	LMU-UNSUP

To probe further: http://aclweb.org/anthology/W18-6401.pdf

Multimodal machine translation tasks

- WMT 2018 example: http://www.statmt.org/wmt18/multimodal-task.html
  - Task 1: task consists of translating English sentences that describe an image into German or French or Czech, given the English sentence itself and the image that it describes (or features from this image)



To probe further: Specia et al. (2016) and Elliott et al. (2017)

WMT 2018 multi-modal task results

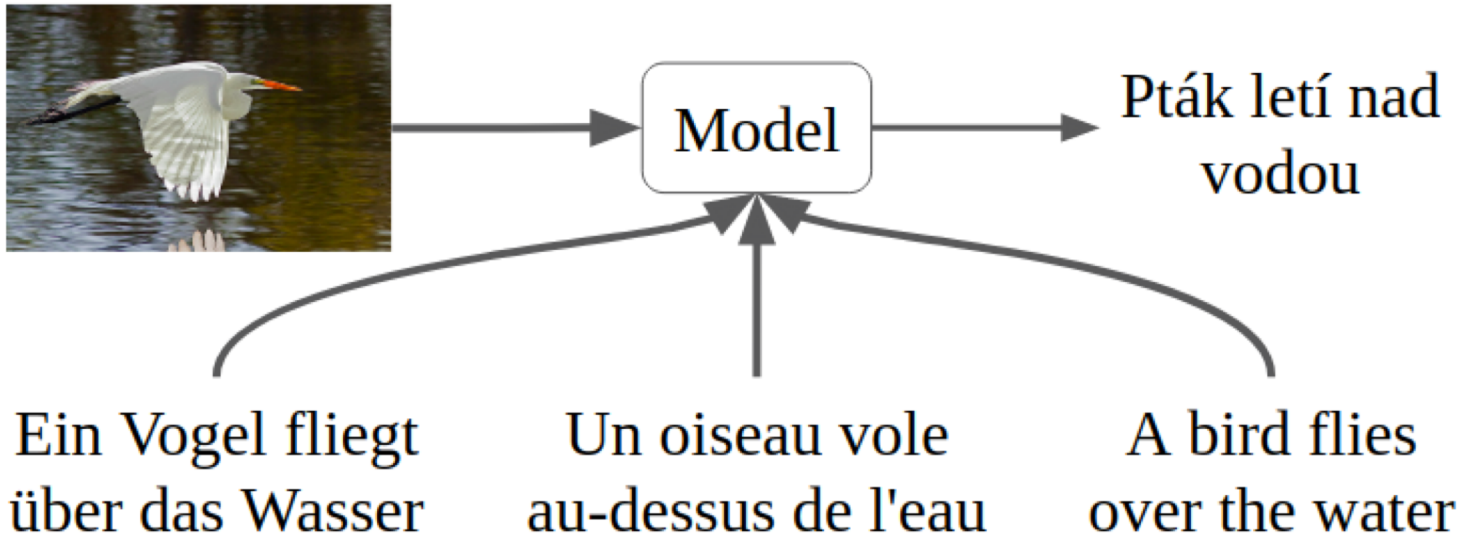
Accuracy metrics
used in sequence modeling

En-De Flickr 2018:

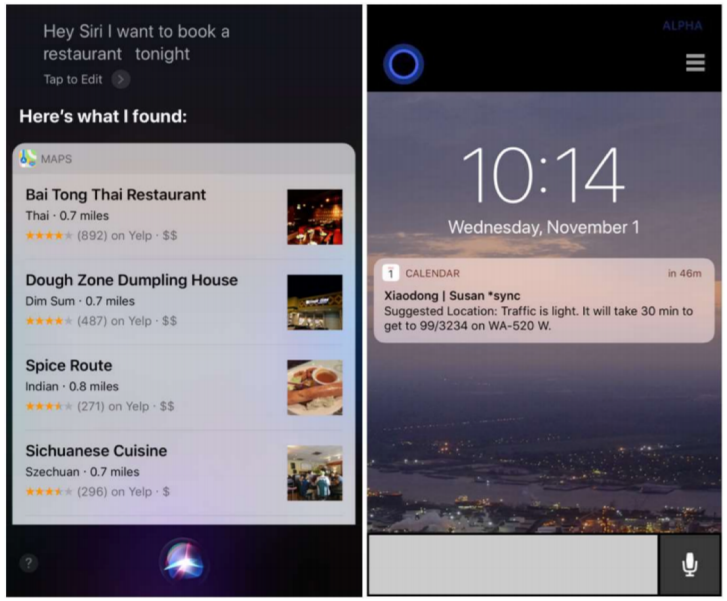
	Submission Name	BLEU	Meteor	TER
1	MeMAD_1_FLICKR_DE_MeMAD-OpenNMT-mmod_U	38.5	56.6	44.6
2	CUNI_1_FLICKR_DE_NeuralMonkeyTextual_U	32.5	52.3	50.8
3	CUNI_1_FLICKR_DE_NeuralMonkeyImagination_U	32.2	51.7	51.7
4	UMONS_1_FLICKR_DE_DeepGru_C	31.1	51.6	53.4
5	LIUMCVC_1_FLICKR_DE_NMTEnsemble_C	31.1	51.5	52.6
6	LIUMCVC_1_FLICKR_DE_MNMTEnsemble_C	31.4	51.4	52.1
7	OSU-BD_1_FLICKR_DE_RLNMT_C	32.3	50.9	49.9
8	OSU-BD_1_FLICKR_DE_RLMIX_C	32.1	50.7	49.6
9	SHEF_1_DE_LT_C	30.5	50.7	53
10	SHEF_1_DE_MLT_C	30.4	50.7	52.9
11	SHEF1_1_DE_ENMT_C	30.9	50.7	52.4
12	SHEF1_1_DE_MFS_C	30.3	50.7	53.1
13	LIUMCVC_1_FLICKR_DE_MNMTSingle_C	28.8	49.9	55.6
14	LIUMCVC_1_FLICKR_DE_NMTSingle_C	29.5	49.9	54.3
15	Baseline	27.6	47.4	55.2
16	AFRL-OHIO-STATE_1_FLICKR_DE_4COMBO_U	24.3	45.4	58.6
17	AFRL-OHIO-STATE_1_FLICKR_DE_2IMPROVE_U	10	25.4	79.2
18	AFRL-OHIO-STATE_1_FLICKR_DE_CAPONLY_U	5	17.7	80.1

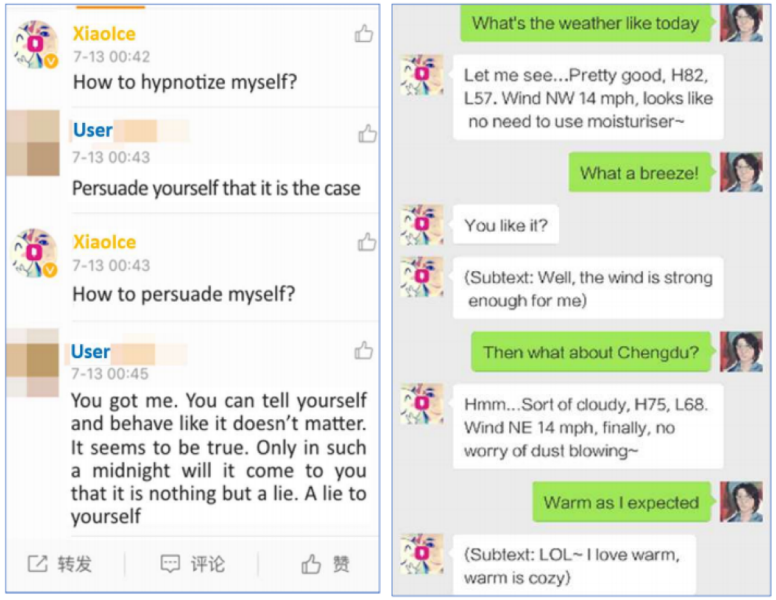
Multisource multimodal machine translation tasks

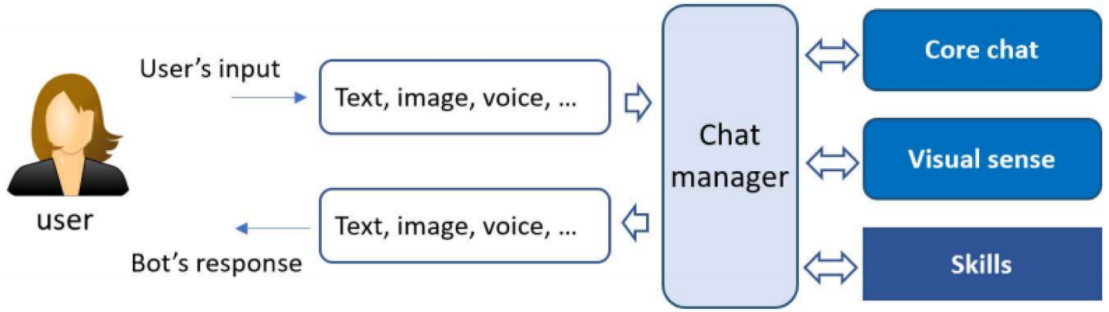
- WMT 2018 task 1b: a new task consisting of translating English sentences that describe an image into Czech, given the English sentence itself, the image that it describes (or features), and parallel sentences in French and German



Virtual assistants, social chatbots







Survey: Shum, He, and Li, From Eliza to XiaoIce: Challenges and Opportunities with Social Chatbots, ArXiv 2018

Some dialogue datasets

- Ubuntu dialogue corpus: a collection of logs from Ubuntu-related chat rooms on the Freenode Internet Relay Chat (IRC) network
  - Common pattern: Q&A convesations
  - https://github.com/rkadlec/ubuntu-ranking-dataset-creator
  - http://dataset.cs.mcgill.ca/ubuntu-corpus-1.0/
  - Lowe et al, paper: https://arxiv.org/pdf/1506.08909.pdf
  - http://www.cs.toronto.edu/~lcharlin/papers/ubuntu_dialogue_dd17.pdf
- Haixun Wang's blog

Amazon Alexa Prize

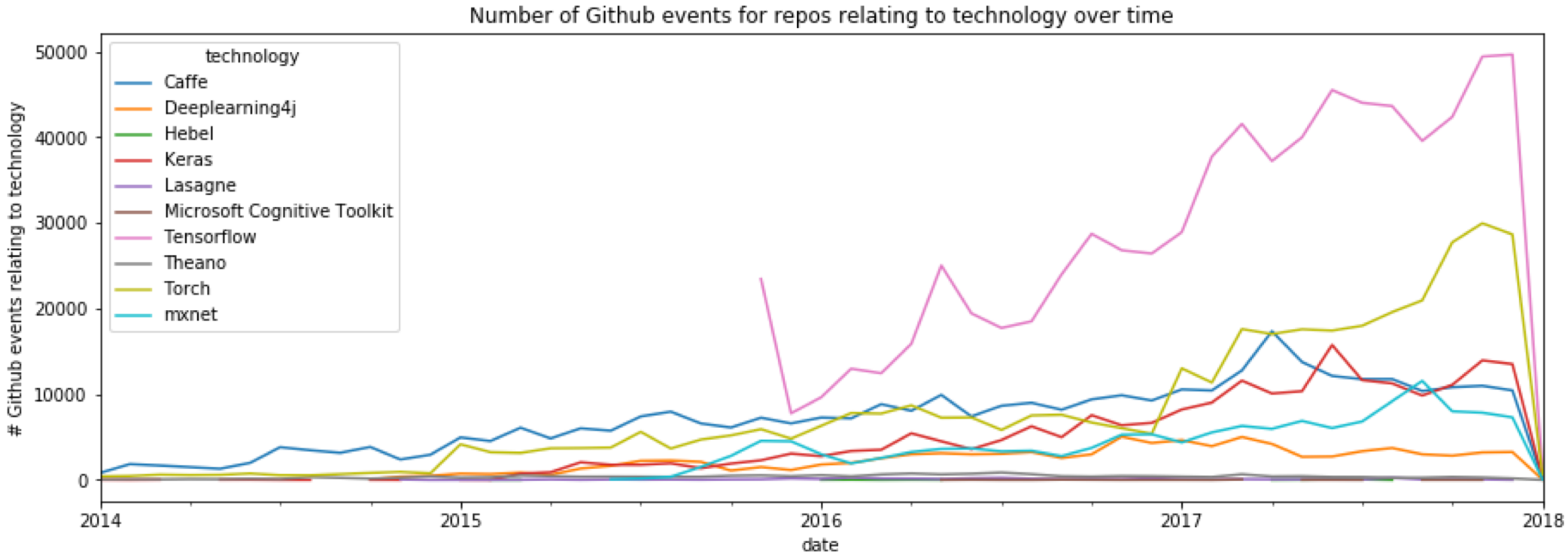
- The Socialbot Challenge: competing teams of college or university students build a conversational Alexa skill using the Alexa Skills Kit APIs that converses with users on popular topics and current events via Amazon Alexa (a “Socialbot”)

- https://developer.amazon.com/alexaprize

Deep learning software frameworks

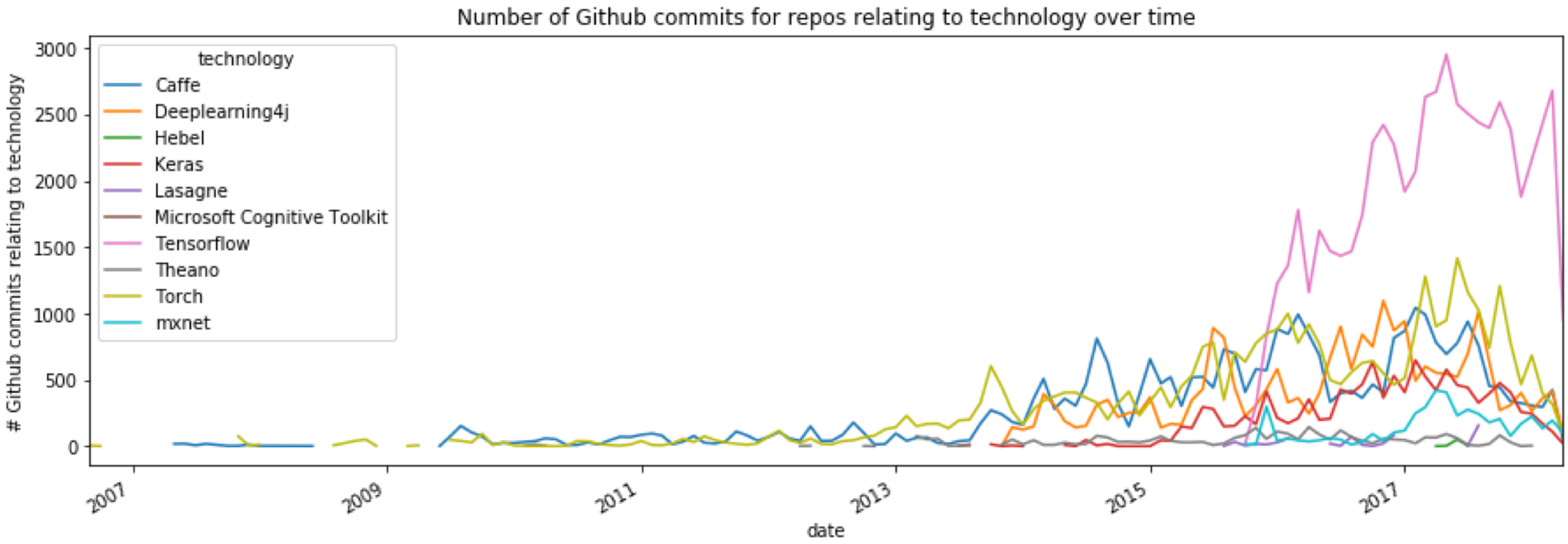
- Software for building and training neural networks
- Several different solutions:
  - Caffe (open source, originally Berkeley)
  - DeepLearning4j (open source, Adam Gibson)
  - Keras (open source, Francois Chollet, Google Engineer)
    - Library for fast experimentation
    - Running on top of TensorFlow, Microsoft Cognitive Toolkit, or Theano
  - Lasagne (lightweight library to build and train in Theano
  - Microsoft Cognitive Toolkit
  - MxNet (open source, supported by several institutions)
  - TensorFlow (open source, originally by Google Brain)
  - Theano (open source, primarily developed by University of Montreal)
  - PyTorch (open source, primarily developed by Facebook and Uber)

Deep learning software frameworks

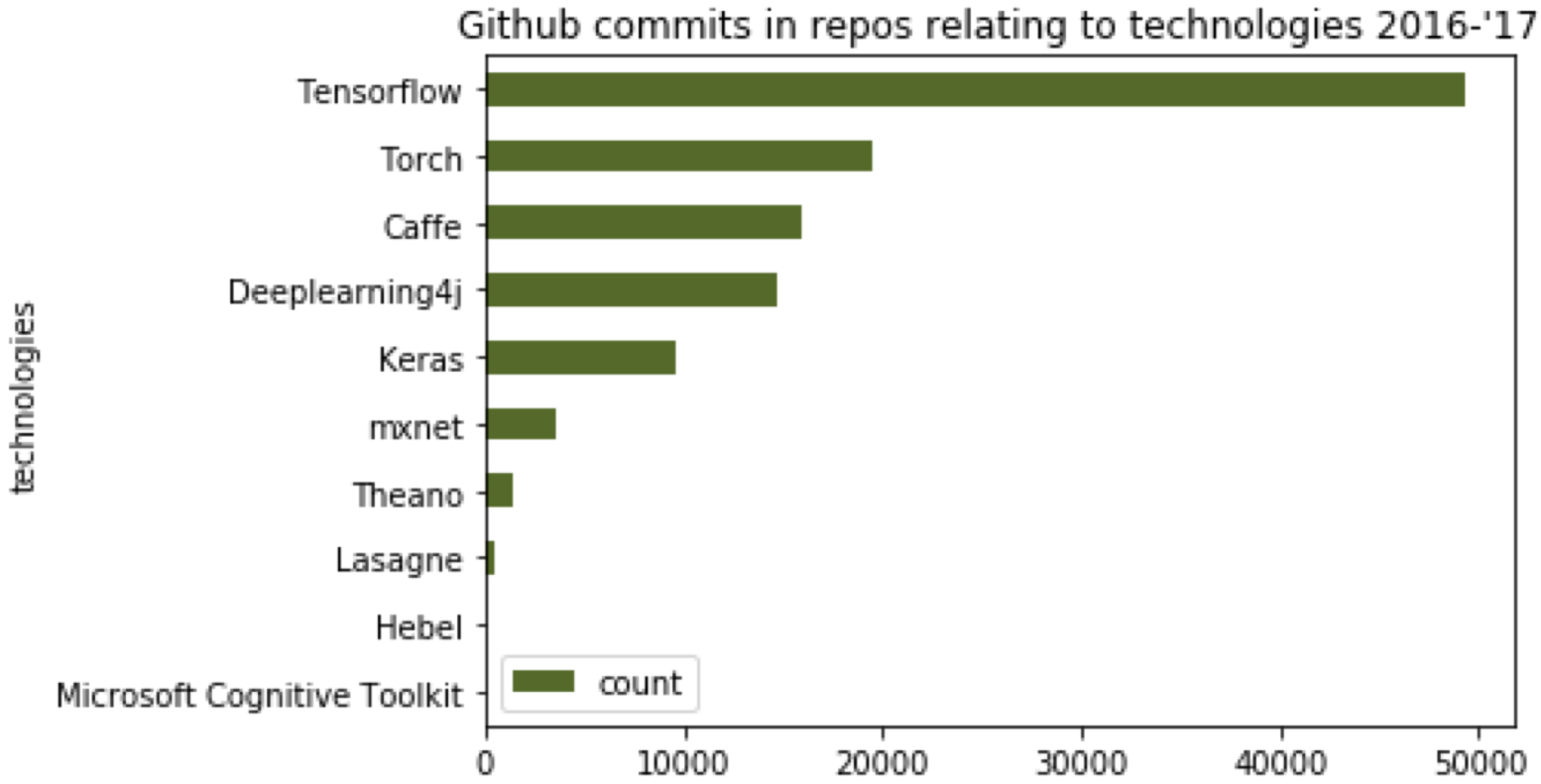


- Credit: Philipp Loick

Deep learning software frameworks (cont'd)

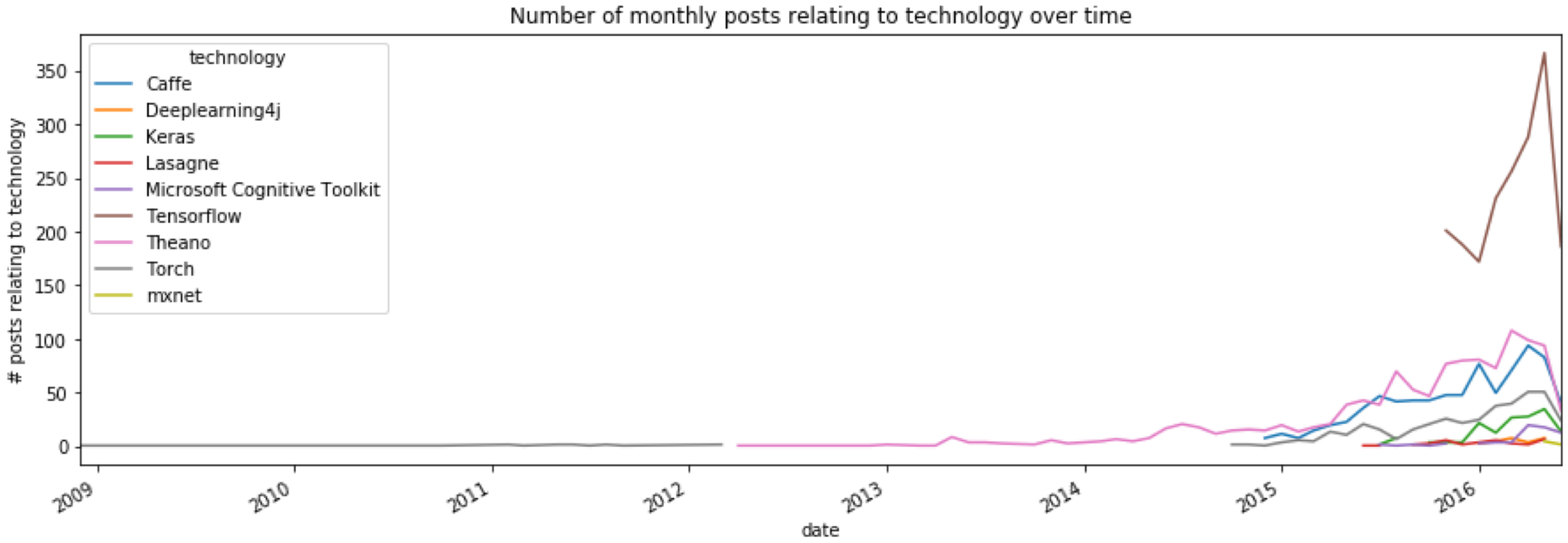


Deep learning software frameworks (cont'd)



Deep learning software frameworks (cont'd)

- StackOverflow activity



Computing systems for training neural networks

- CPU (Central Processor Unit)
  - Your laptop, cloud virtual machines
- GPU (Graphical Processor Unit)
  - NVIDIA's GPUs: Tesla K80, M60, P40, P100, V100
  - Colab 1 GPU for free (NVIDIA Tesla K80)
- TPU (Tensor Processor Unit)
  - Google's custom-developed application-specific integrated circuits (ASICs) used to accelerate machine learning workloads
  - https://cloud.google.com/tpu
- FPGA (Field Programmable Gate Arrays)
  - Microsoft's integrated circuit that can be programmed in the filed
  - Launched on Azure in 2018 (preview)
  - Mark Russinovich's MBSBuild talk

An example GPU (NVIDIA)

	GeForce RTX 2070 Founders Edition	GeForce RTX 2070
GPU Architecture	Turing	Turing
RTX-OPS	45T	42T
Boost Clock	1710 MHz (OC)	1620 MHz
Frame Buffer	8 GB GDDR6	8 GB GDDR6
Memory Speed	14 Gbps	14 Gbps

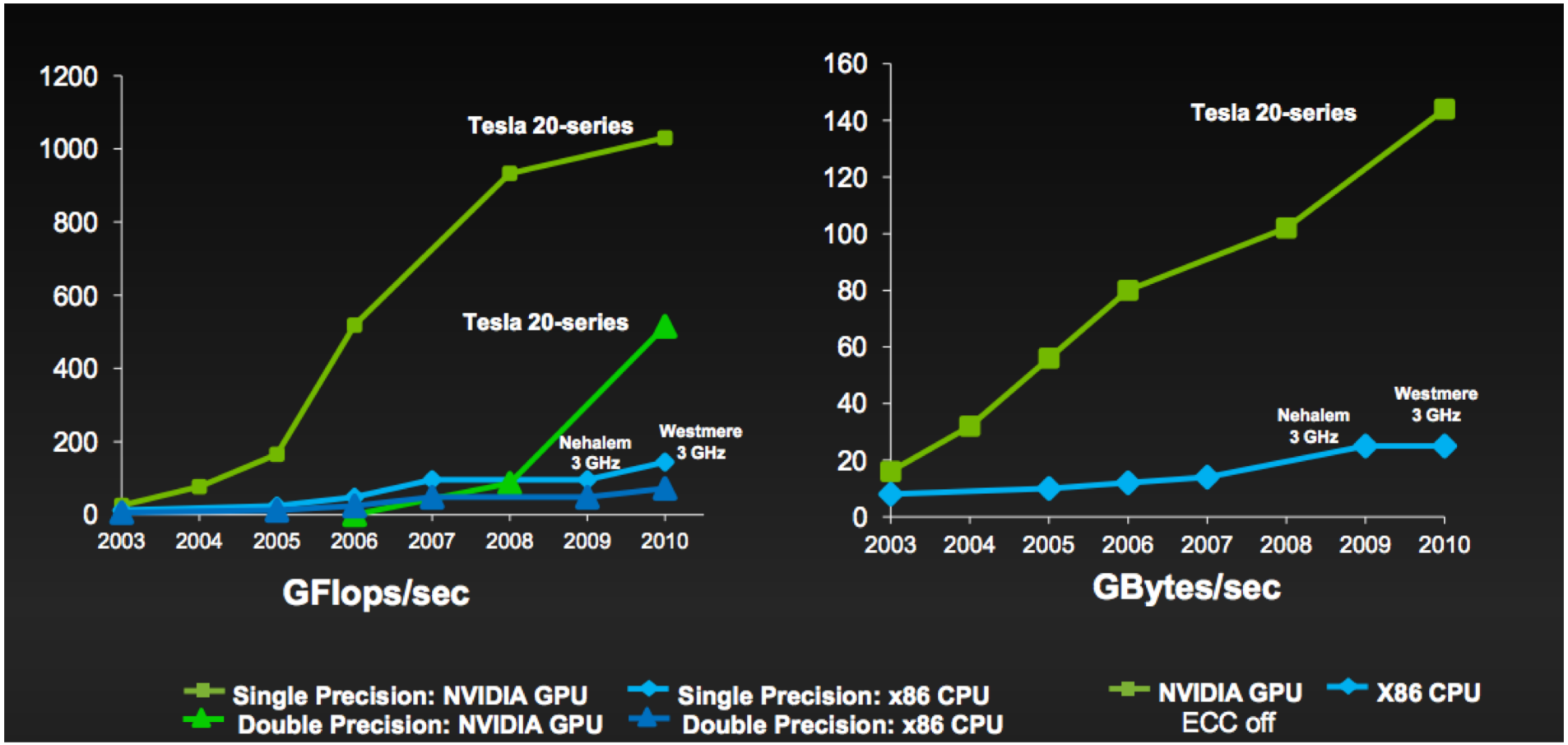
https://www.nvidia.com

CPU vs GPU computing

- CPU computing
  - Optimized for low-latency access to caches data sets
  - Control logic for out-of-order and speculative execution

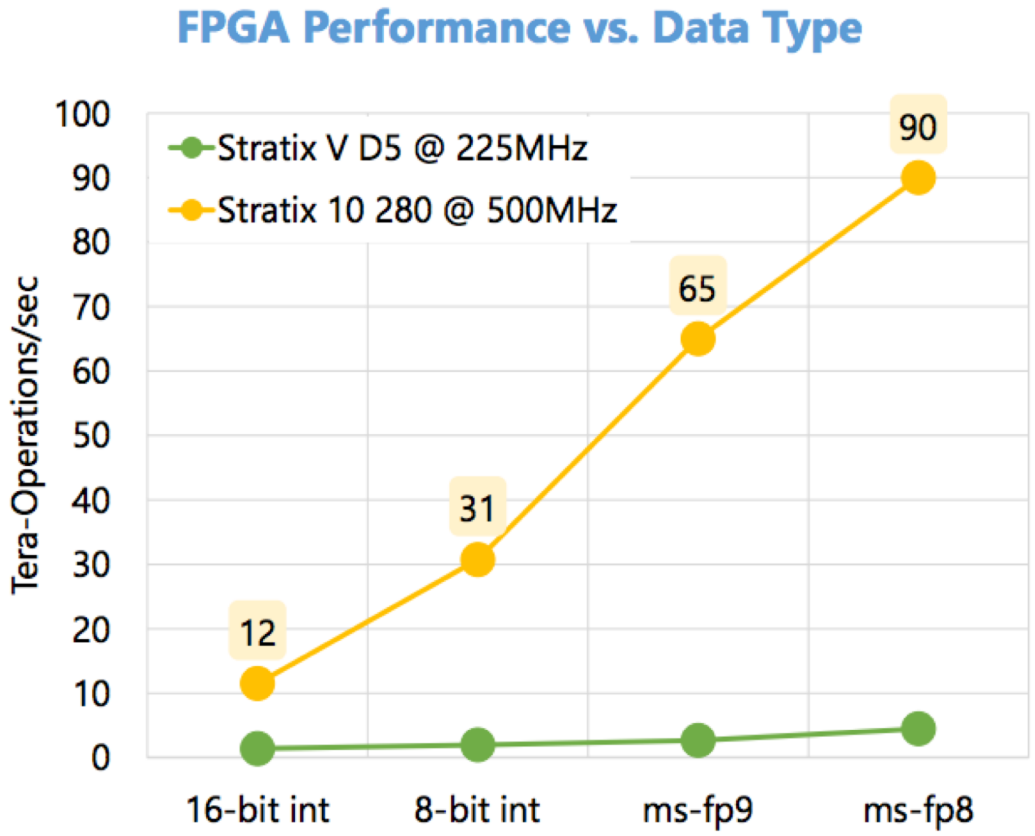
- GPU computing
  - Optimized for data-parallel throughput computation
  - Architecture tolerant of memory latency
  - More transistors dedicated to computation

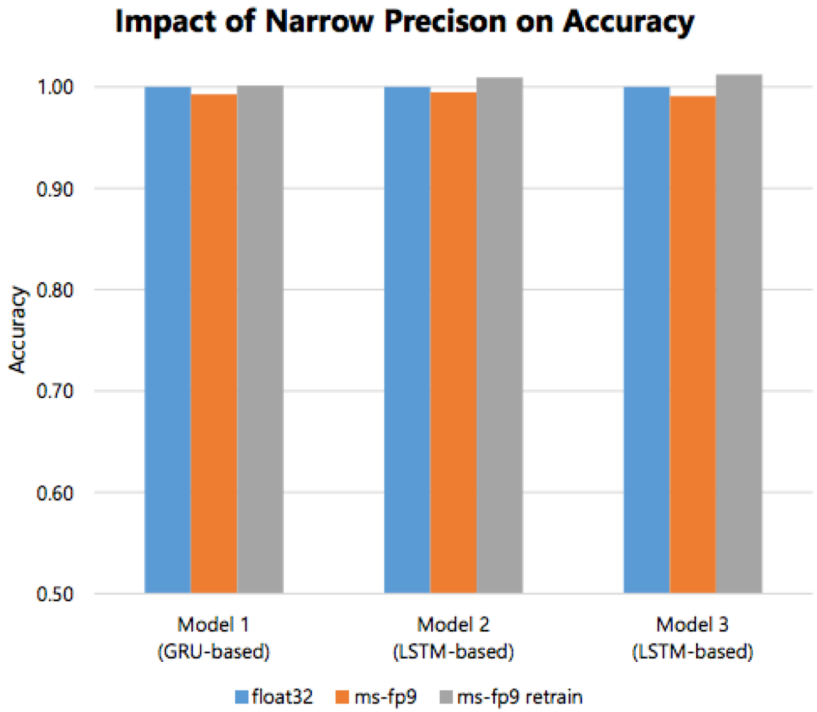
CPU vs GPU computing performance



- Source: http://www.int.washington.edu/PROGRAMS/12-2c/week3/clark_01.pdf

Speedup by precision truncation (FPGA example)





- Chung et al, Accelerating persistent neural networks at datacenter scale, Microsoft, 2017

When to use different processing units?

- CPUs
  - Quick prototyping that requires maximum flexibility
  - Simple models that do not take long to train
  - Small models with small effective batch sizes
  - Models that are dominated by custom TensorFlow operations written in C++
  - Models that are limited by available I/O or the networking bandwidth of the host system
- GPUs
  - Models that are not written in TensorFlow or cannot be written in TensorFlow
  - Models for which source does not exist or is too onerous to change
  - Models with a significant number of custom TensorFlow operations that must run at least partially on CPUs
  - Models with TensorFlow ops that are not available on Cloud TPU (available TensorFlow ops)
  - Medium-to-large models with larger effective batch sizes
- TPUs
  - Models dominated by matrix computations
  - Models with no custom TensorFlow operations inside the main training loop
  - Models that train for weeks or months
  - Larger and very large models with very large effective batch sizes

Source: https://cloud.google.com/tpu/docs/tpus

MLPerf

- A broad ML benchmark suite for measuring performance of ML software frameworks, ML hardware accelerators, and ML cloud platforms
  - https://mlperf.org/
- MLPerf v.05 results:

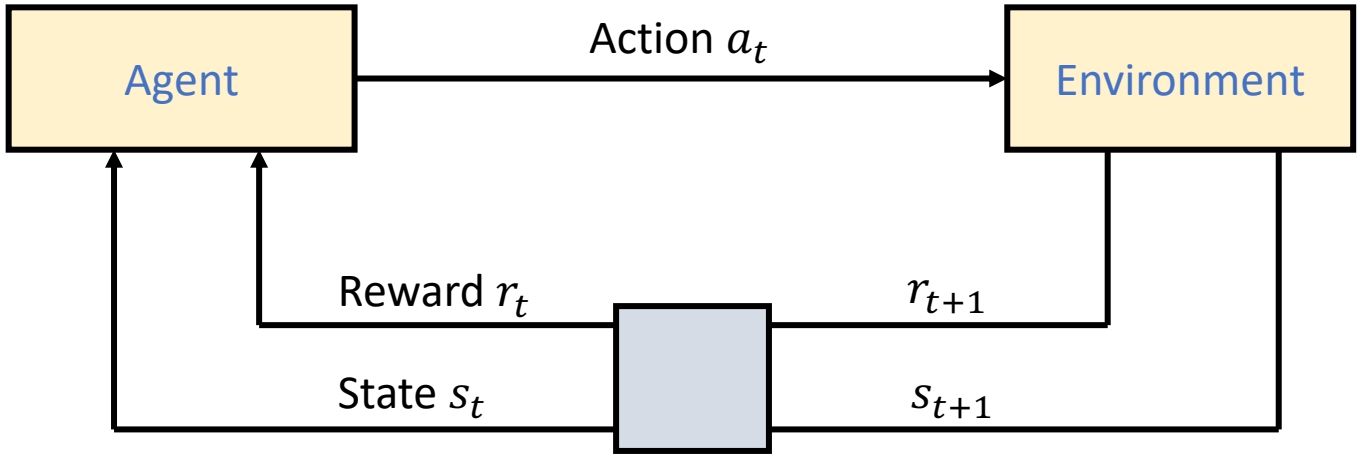
Closed Division Speedups													
						Benchmark results (speedup relative to reference implementation)							
						Image classifi-cation	Object detection, light-weight	Object detection, heavy-wt.	Translation , recurrent	Translation , non-recur.	Recom-mendation	Reinforce-ment Learning	
						ImageNet	COCO	COCO	WMT E-G	WMT E-G	MovieLens-20M	Pro games	
#	Submitter	Hardware	Chip count and type		Software	ResNet-50 v1.5	SSD w/ ResNet-34	Mask-R-CNN	NMT	Transformer	NCF	Mini Go	Cloud Scale
Available in cloud													
1	Reference	Pascal P100	1	a	Unoptimized reference	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	Google	TPUv2.8	4	a	TF 1.12	29.3	8.5		28.1				2.6
3		TPUv2.512 + TPUv2.8	260	a	TF 1.12	781.5							171.6
4		TPUv3.8	4	a	TF 1.12	48.2	11.1		43.1				4.2
5		8x Volta V100	8	a	TF 1.12, cuDNN 7.4	64.1							11.4

Computing systems references

- NVIDIA
  - https://www.nvidia.com/en-us/deep-learning-ai
- GPU cloud computing
  - Google Compute Engine
  - Amazon EC instances
  - Microsoft Azure
  - NVIDIA GPU cloud
- TPU cloud computing
  - Google Cloud Tensor Processing Units
- FPGA
  - Microsoft research project Brainwave

Reinforcement learning problem

- Reinforcement learning problem: an agent learning from interaction in an environment aiming to maximize a long-term return



Atari game example

score

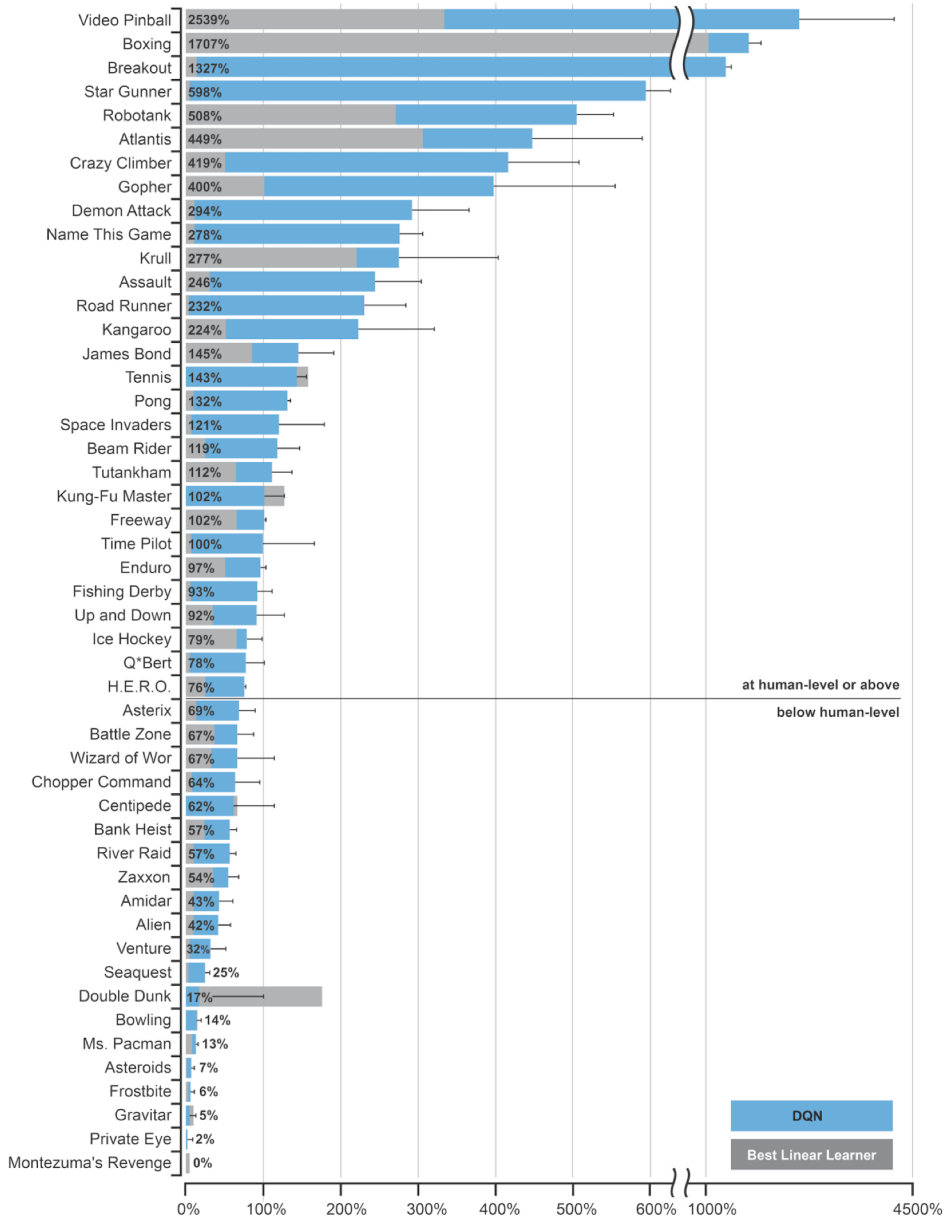
number of lives

# of players

Video: https://youtu.be/TmPfTpjtdgg

Atari game performance

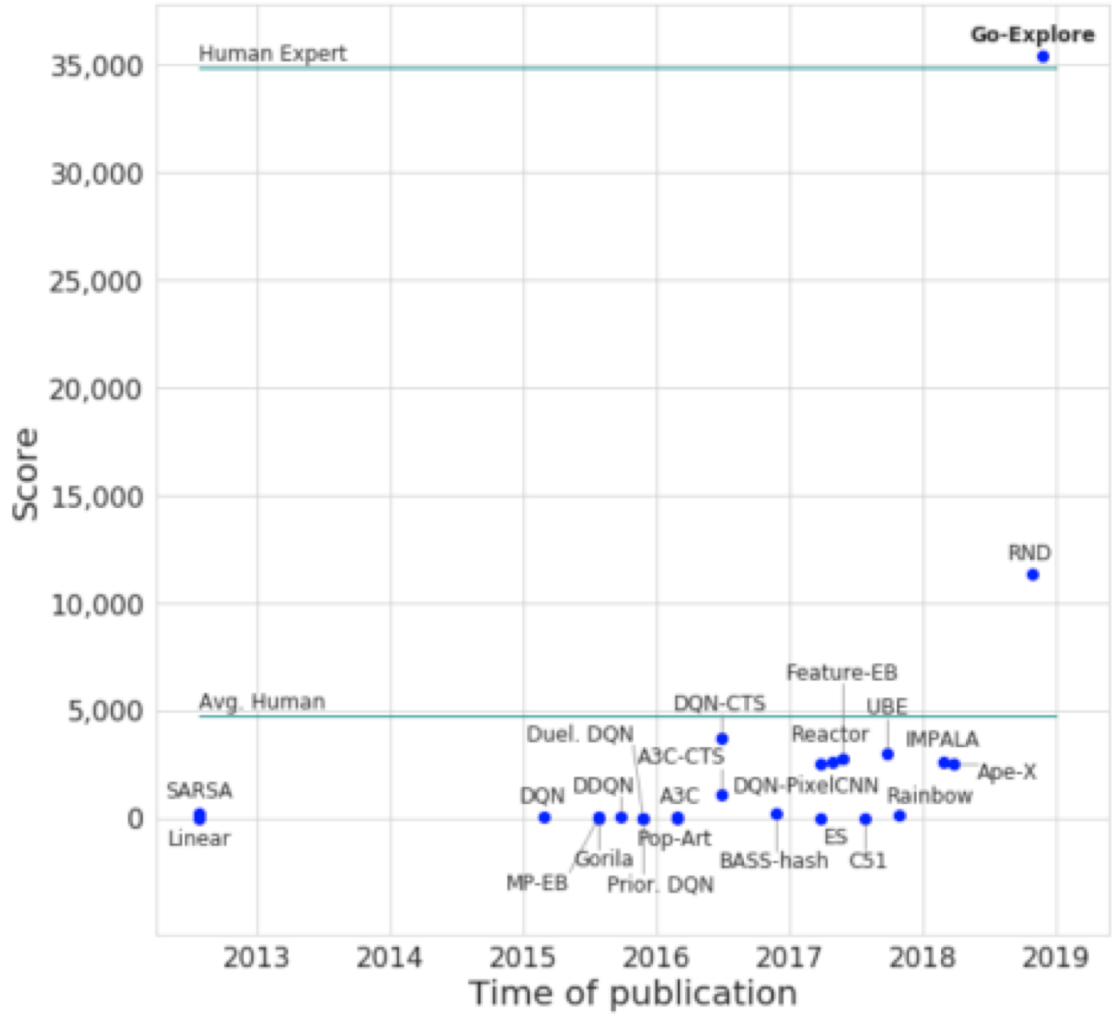
- Reinforcement learning with action-value function approximation
- Linear: each state represented by a feature vector, linear function approximation
- DQN: non-linear function approximation using a deep neural network



- Mnih et al, Human level control through deep reinforcement learning, Nature 2015

Atari game Montezuma's Revenge

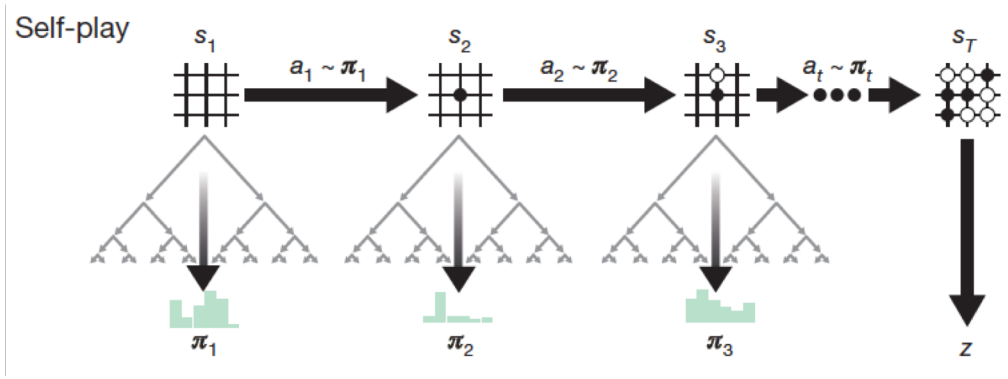
- Hard exploration problem: agent has to learn a complex task with infrequent and deceptive feedback

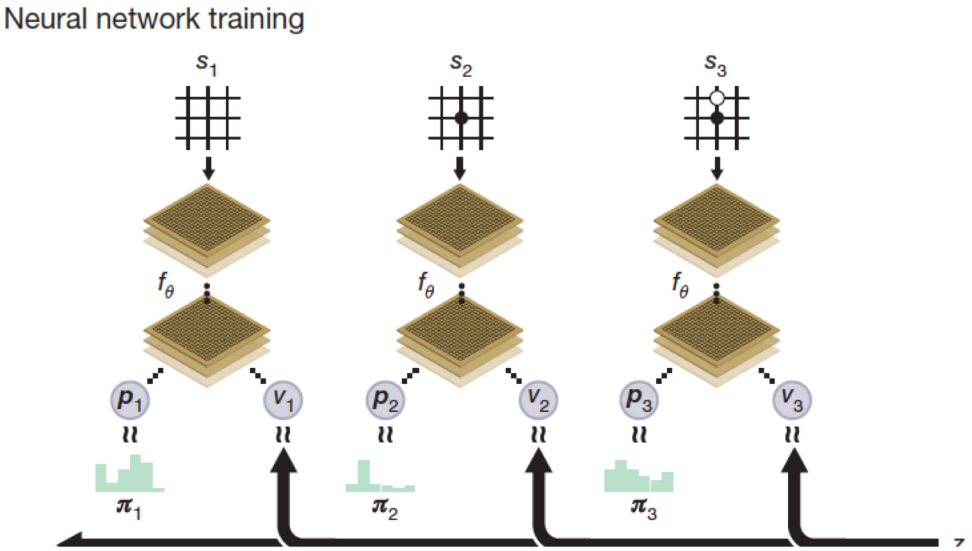


- Uber blog: https://eng.uber.com/go-explore/

AlphaGo

4-1 victory against Lee Sedol, widely considered to be the greatest player of the past decade





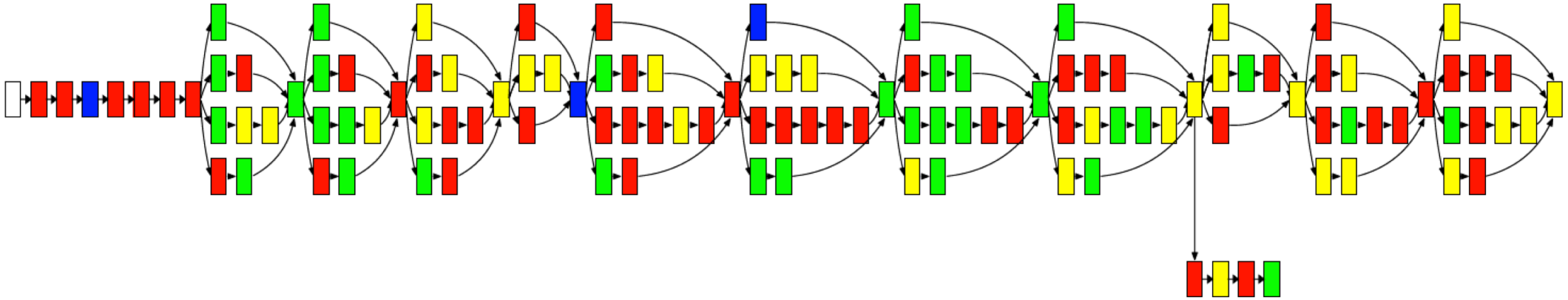
- Silver et al, Mastering the game of Go without human knowledge, Nature 2017

Project Malmo

- https://www.microsoft.com/en-us/research/project/project-malmo/
  - AI experimentation platform built on top of Minecraft, designed to support fundamental research in artificial intelligence

- MARLO 2018 challenge https://www.crowdai.org/challenges/marlo-2018
  - Multi-agent reinforcement learning in Minecraft

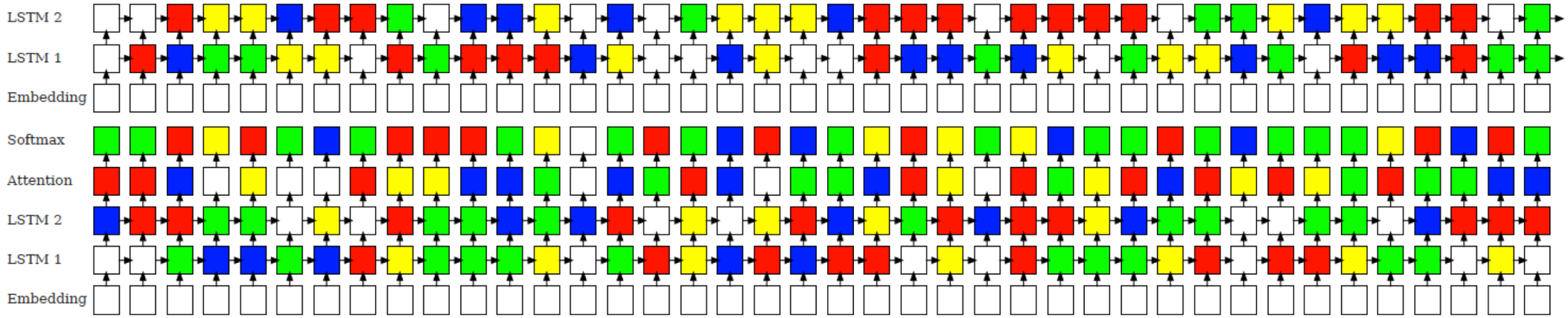
Reinforcement learning for resource allocation



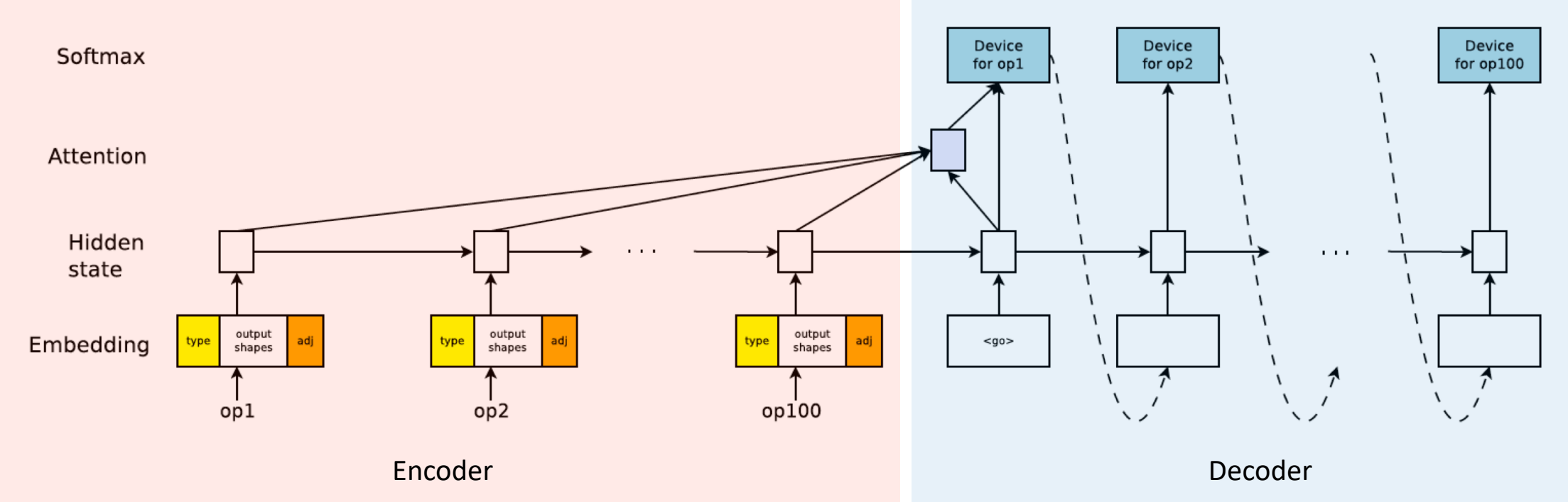
- RL based device placement for Inception-V3 neural network
  - GPU devices are denoted by colors
  - CPU devices are denoted by transparent color

- Mirhoseini et al, Device placement optimization with reinforcement learning, ICML 2017

Cont’d: for a neural machine translation network



Device placement network architecture



- A sequence to sequence model following an encoder-decoder architecture
- Training with policy gradient method

Running time performance

Tasks	Single-CPU	Single-GPU	#GPUs	Scotch	MinCut	Expert	RL-based	Speedup
RNNLM (batch 64)	6.89	**1.57**	2	13.43	11.94	3.81	**1.57**	0.0%
			4	11.52	10.44	4.46	**1.57**	0.0%
NMT (batch 64)	10.72	OOM	2	14.19	11.54	4.99	**4.04**	23.5%
			4	11.23	11.78	4.73	**3.92**	20.6%
Inception-V3 (batch 32)	26.21	**4.60**	2	25.24	22.88	11.22	**4.60**	0.0%
			4	23.41	24.52	10.65	**3.85**	19.0%

Machine learning conferences

- Conference on Neural Information Processing Systems (NeurIPS)
  - NeurIPS 2018
  - Previously called NIPS
- International Conference on Machine Learning (ICML)
  - ICML 2019
- Conference on Learning Theory (COLT)
  - COLT 2019
- International Conference on Learning Representations
  - ICLR 2019

Speech recognition conferences

- IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
  - ICASSP 2018
- INTERSPEECH
  - INTERSPEECH 2018
- SIGdial: Special Interest Group on Discourse and Dialogue
  - SIGdial

Neural machine translation conferences

- Annual Meeting of the Association for Computational Linguistics (ACL)
  - ACL 2018
- Empirical Methods in Natural Language Processing (EMNLP)
  - EMNLP 2018
- Conference on Machine Translation (WMT)
  - WMT 2018
  - Previously Workshop on Statistical Machine Translation

Information for Seminar class 1

- Seminar class 1: getting started with TensorFlow
  - Introduction to TensorFlow
  - Introduction to computational graphs
  - Basic operations in TensorFlow
  - Graph building and execution
  - Example: linear regression

- **To do before the seminar class**:
  - Install TensorFlow using instructions given here:

    https://github.com/lse-st449/lectures/blob/master/Week01/Class/SetupTensorFlow.md