Deep Supervised Learning of Representations

Yoshua Bengio

July 4, 2016

Deep Learning Workshop

IDIAP

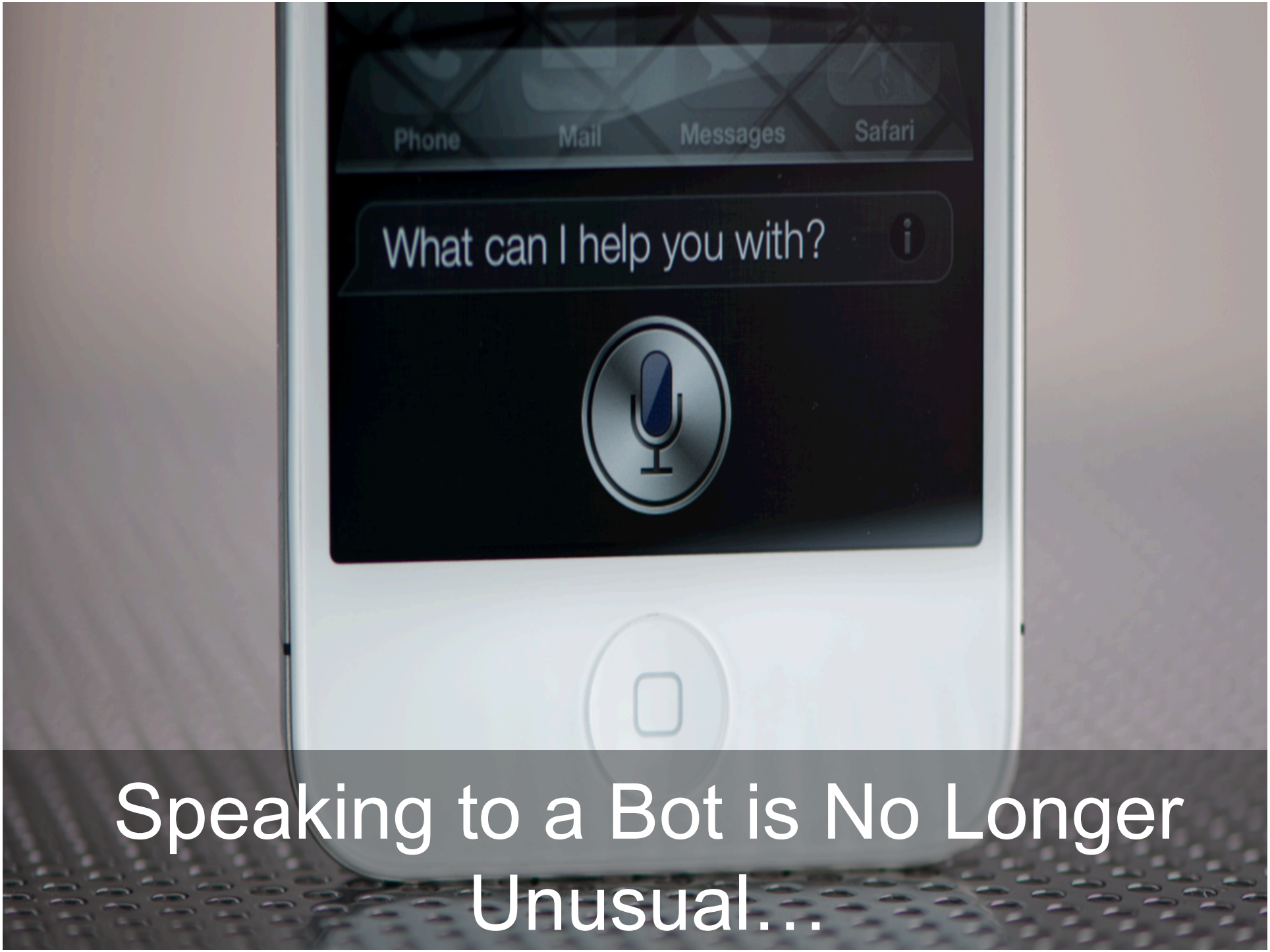
PLUG: **Deep Learning**, MIT Press book in press.
Chapters will stay online.

MILA

Université de Montréal

Cars are now driving themselves…

(far from perfectly, though)



Speaking to a Bot is No Longer Unusual…


March 2016:
World Go Champion
Beaten by Machine

A new revolution seems to be in the work after the industrial revolution.

Devices are becoming intelligent.

And Deep Learning is at the epicenter of this revolution.

IT Companies are Racing into Deep Learning

AI Breakthrough

- **Deep Learning: machine learning algorithms based on learning multiple levels of representation / abstraction.**

Amazing improvements in error rate in object recognition, object detection, speech recognition, and more recently, in natural language processing / understanding

Breakthrough in deep learning

A Canadian-led trio at CIFAR initiated the deep learning AI revolution

- Fundamental breakthrough in 2006:

first successful recipe for training a deep supervised neural network

- Second major advance in 2011, with rectifiers
- Breakthroughs in applications since then, especially the AlexNet 2012.

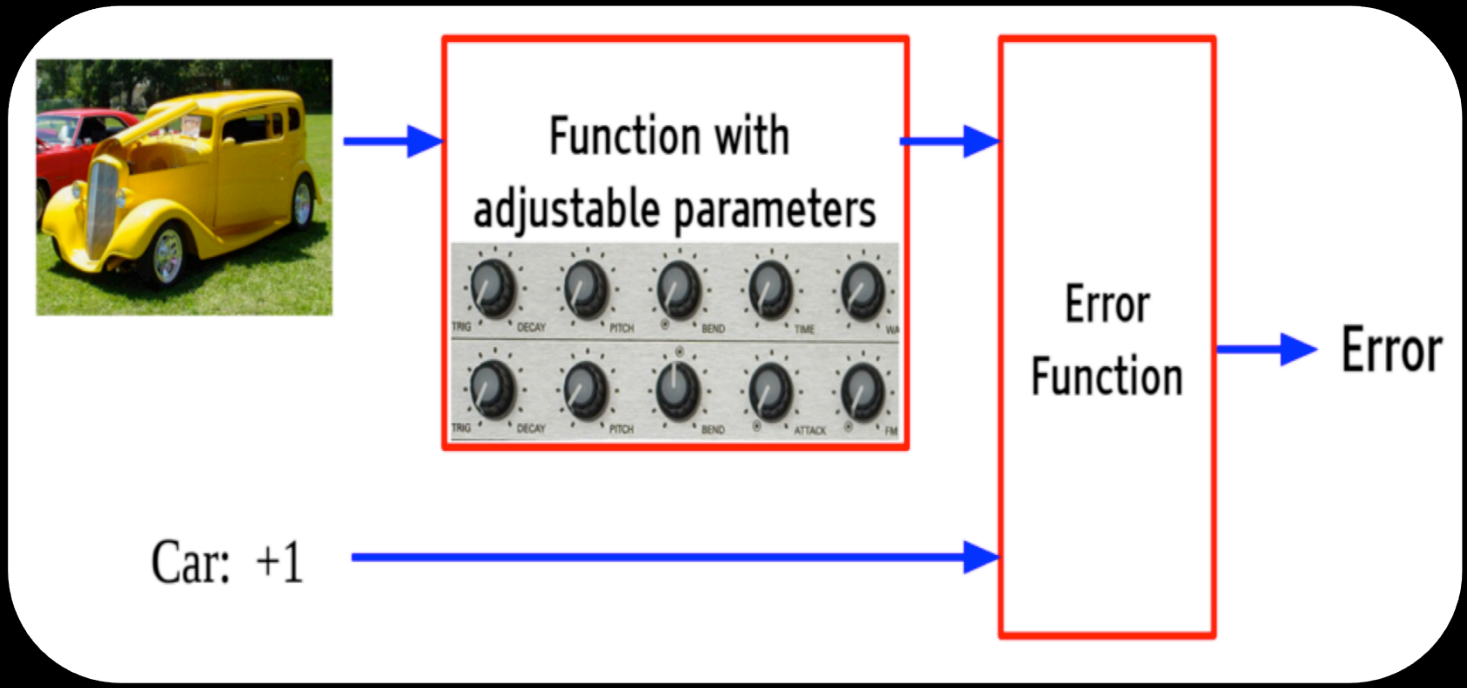
AI Needs Knowledge

- Failure of classical AI: a lot of knowledge is not formalized, expressed with words
- Solution: computer gets knowledge from data, learns from examples

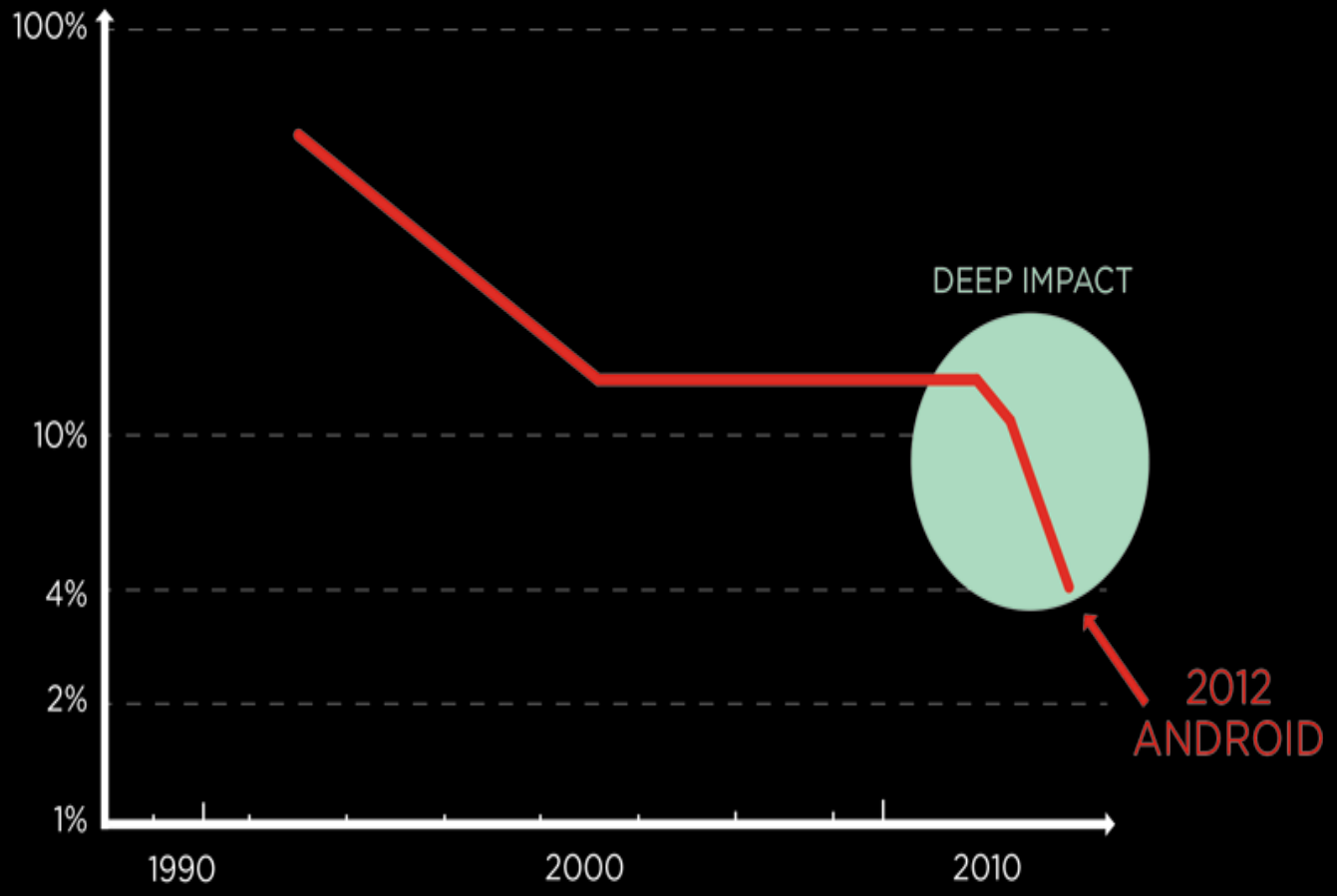
MACHINE LEARNING

Learning by gradually optimizing parameters to better fit examples

Learning agent sees examples and tries to fit them, by gradually optimizing some internal parameters, being trained to perform task



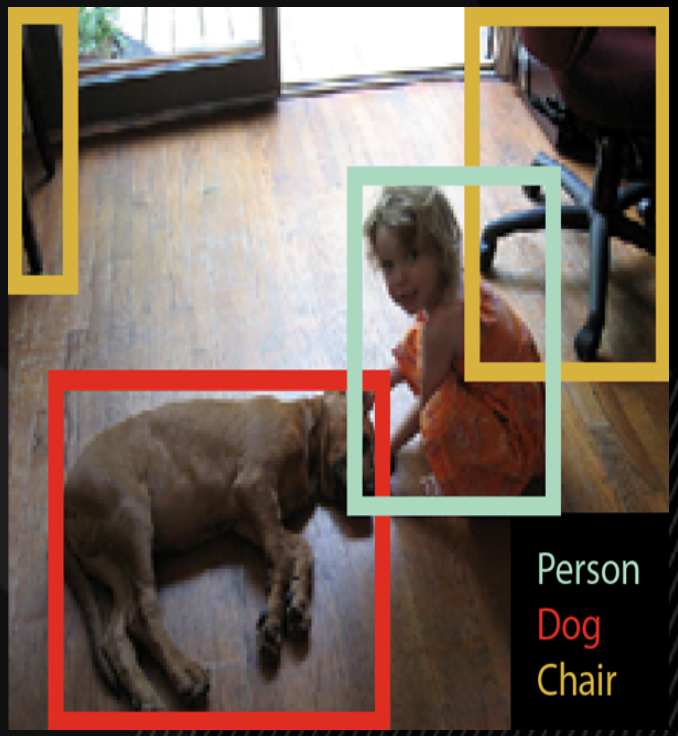
2010-2012: breakthrough in speech recognition



Source: Microsoft

2012-2015: breakthrough in computer vision

- Graphics Processing Units (GPUs) + 10x more data
- 1,000 object categories,
- Facebook: millions of faces
- ***2015: human-level performance***



From computer vision to self-driving cars: 2016

Holmdel, New Jersey
February 2016

At the Heart of Deep Learning: Backprop

Back-Prop

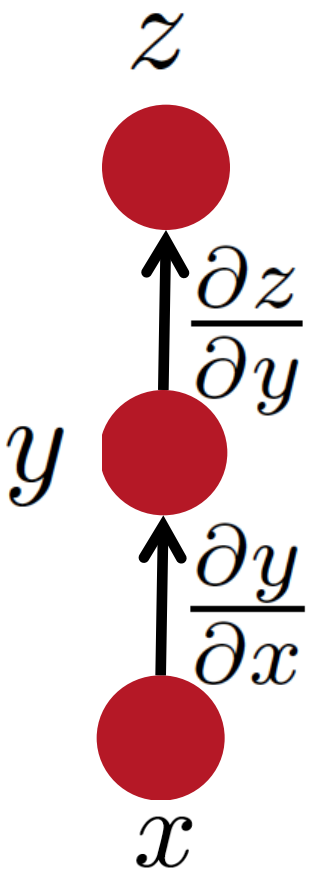
- Compute gradient of example-wise loss wrt parameters

- Simply applying the derivative chain rule wisely

$$z = f(y) \quad y = g(x) \quad \frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$$

- *If computing the loss(example, parameters) is O(n) computation, then so is computing the gradient*

Simple Chain Rule



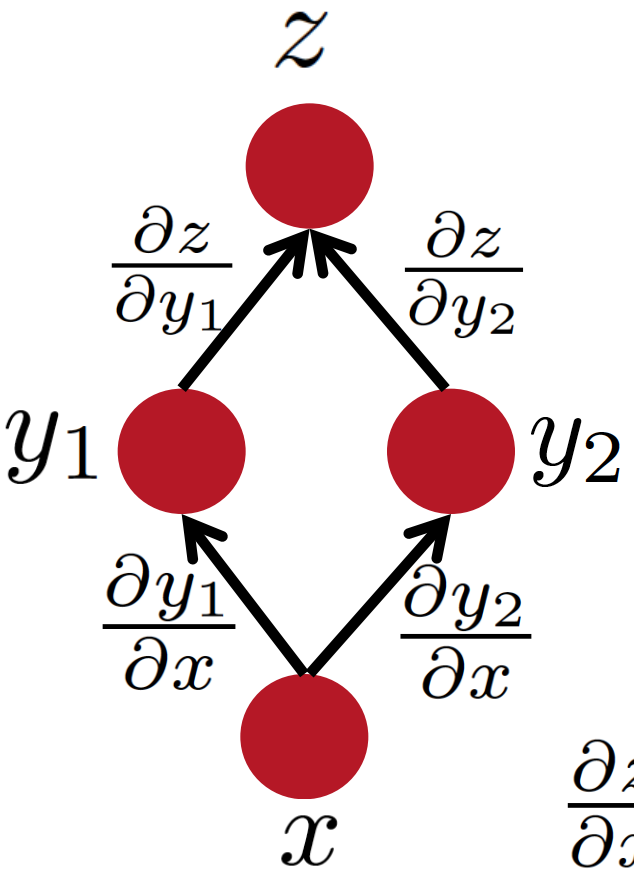
$$\Delta z = \frac{\partial z}{\partial y}\Delta y$$

$$\Delta y = \frac{\partial y}{\partial x}\Delta x$$

$$\Delta z = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\Delta x$$

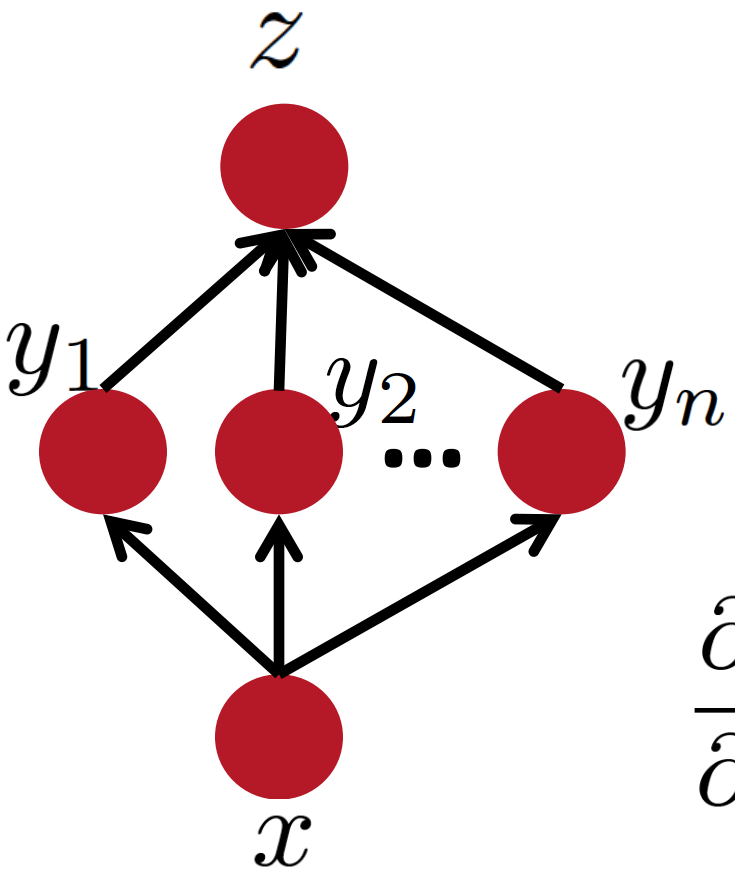
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$$

Multiple Paths Chain Rule



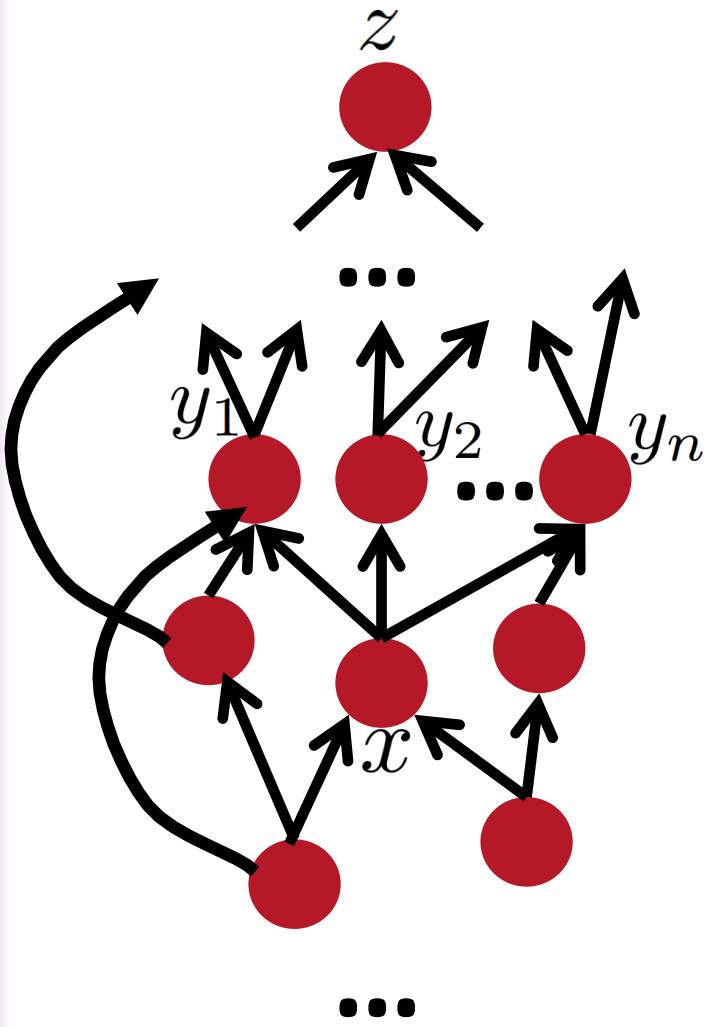
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y_1}\frac{\partial y_1}{\partial x} + \frac{\partial z}{\partial y_2}\frac{\partial y_2}{\partial x}$$

Multiple Paths Chain Rule - General



$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

Chain Rule in Flow Graph

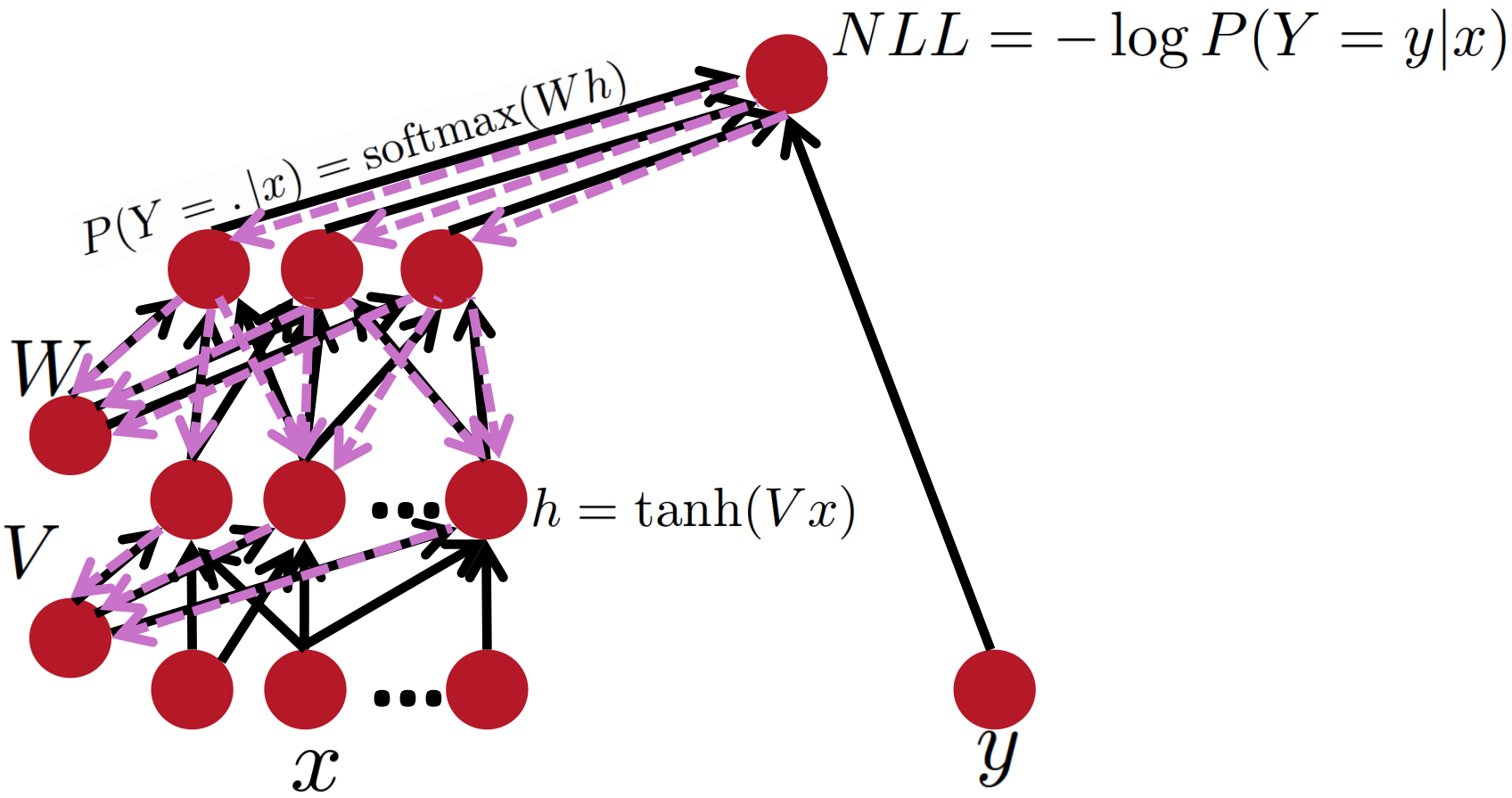


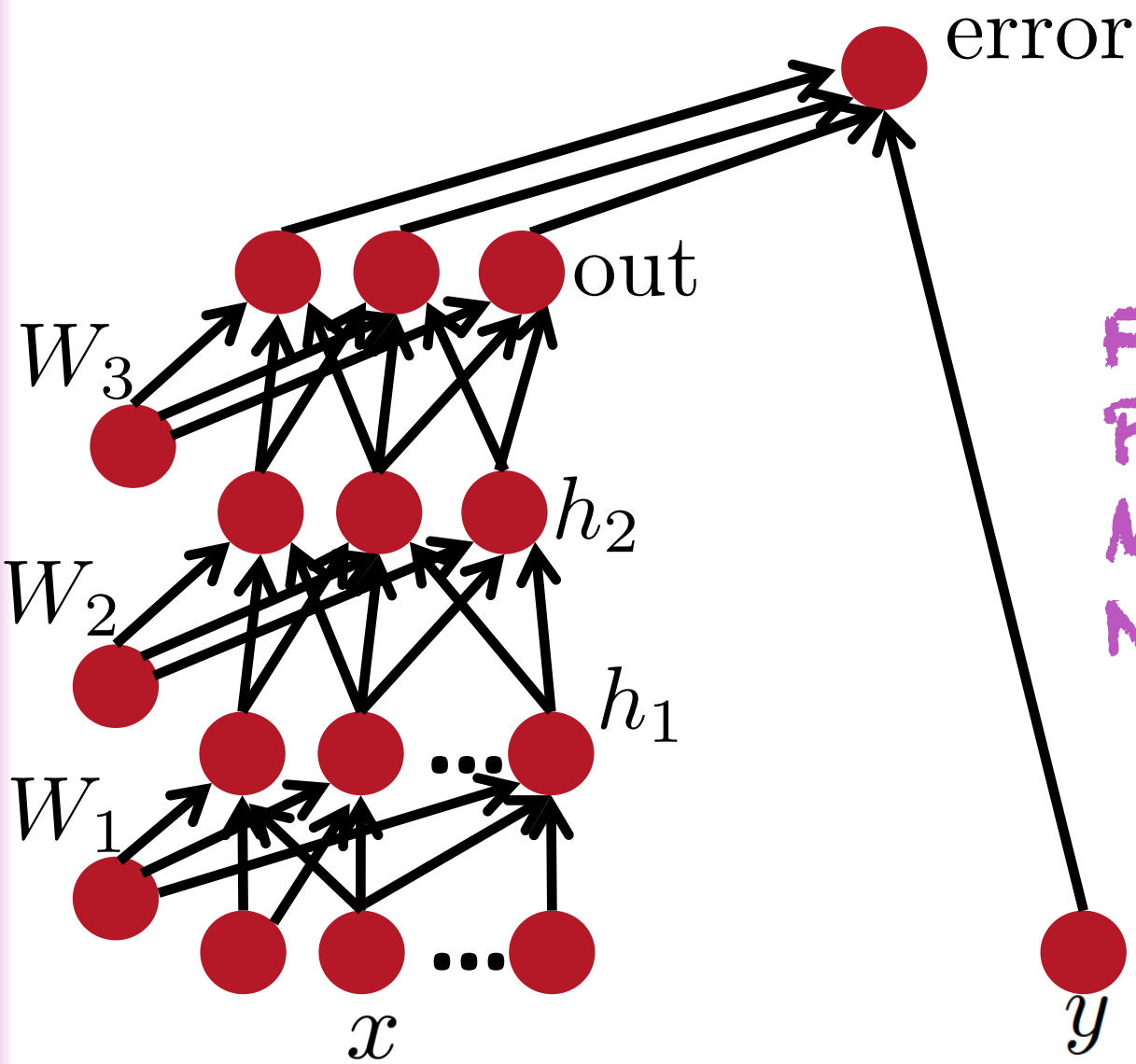
Flow graph: any directed acyclic graph
node = computation result
arc = computation dependency

$\{y_1, y_2, \ldots y_n\}$ = successors of x

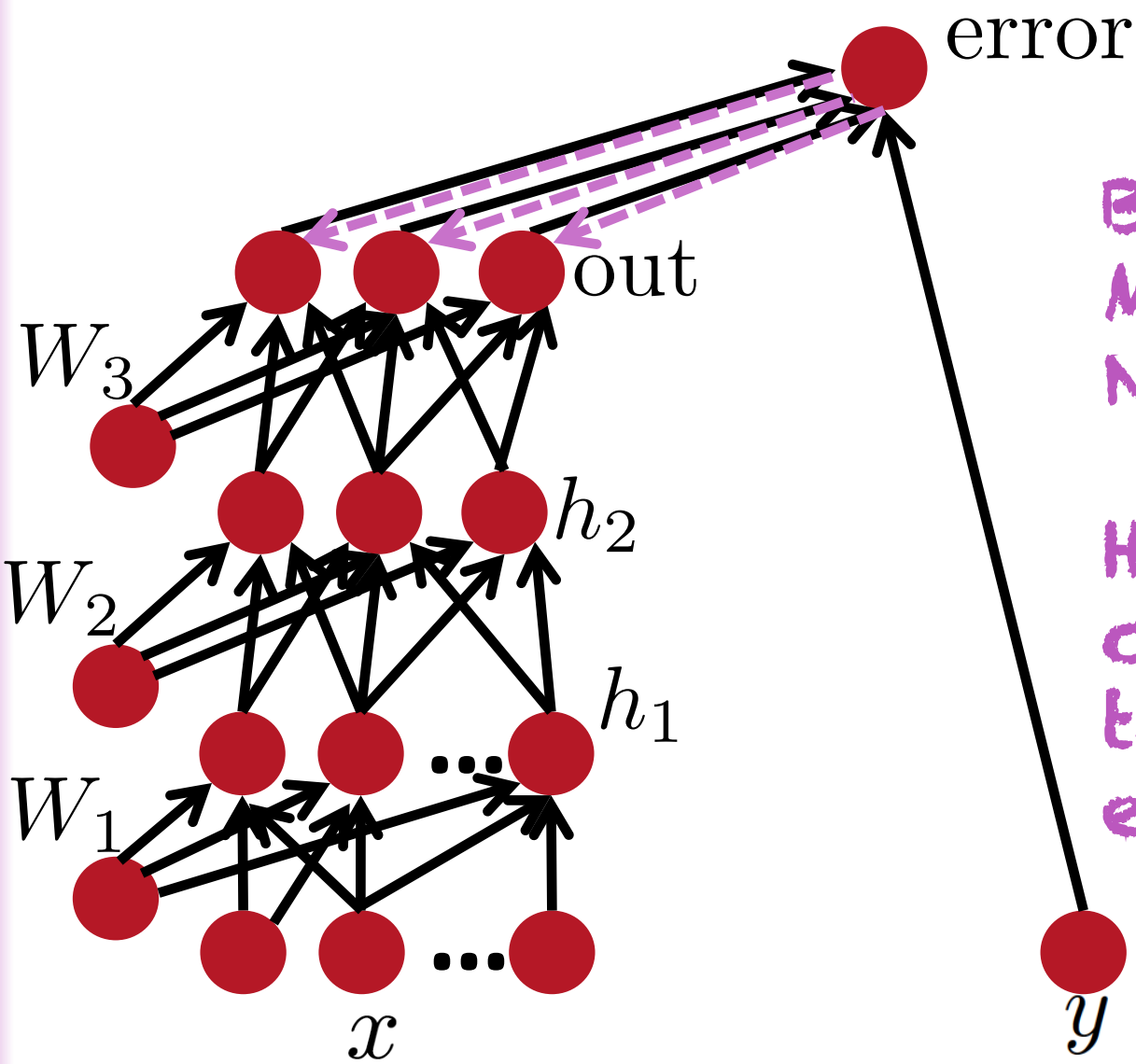
$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

Back-Prop in Multi-Layer Net



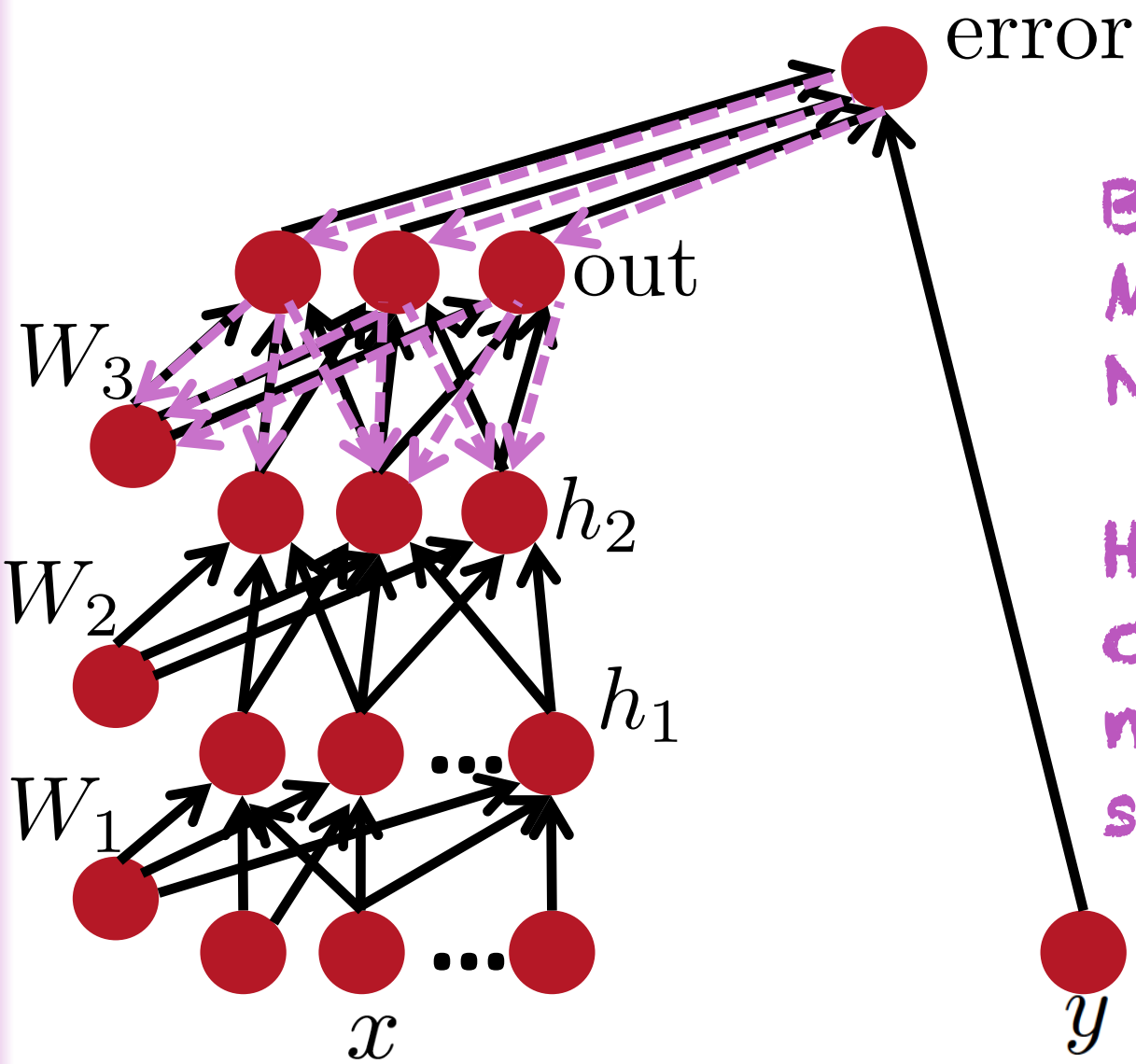


Forward-Prop in Multi-Layer Net



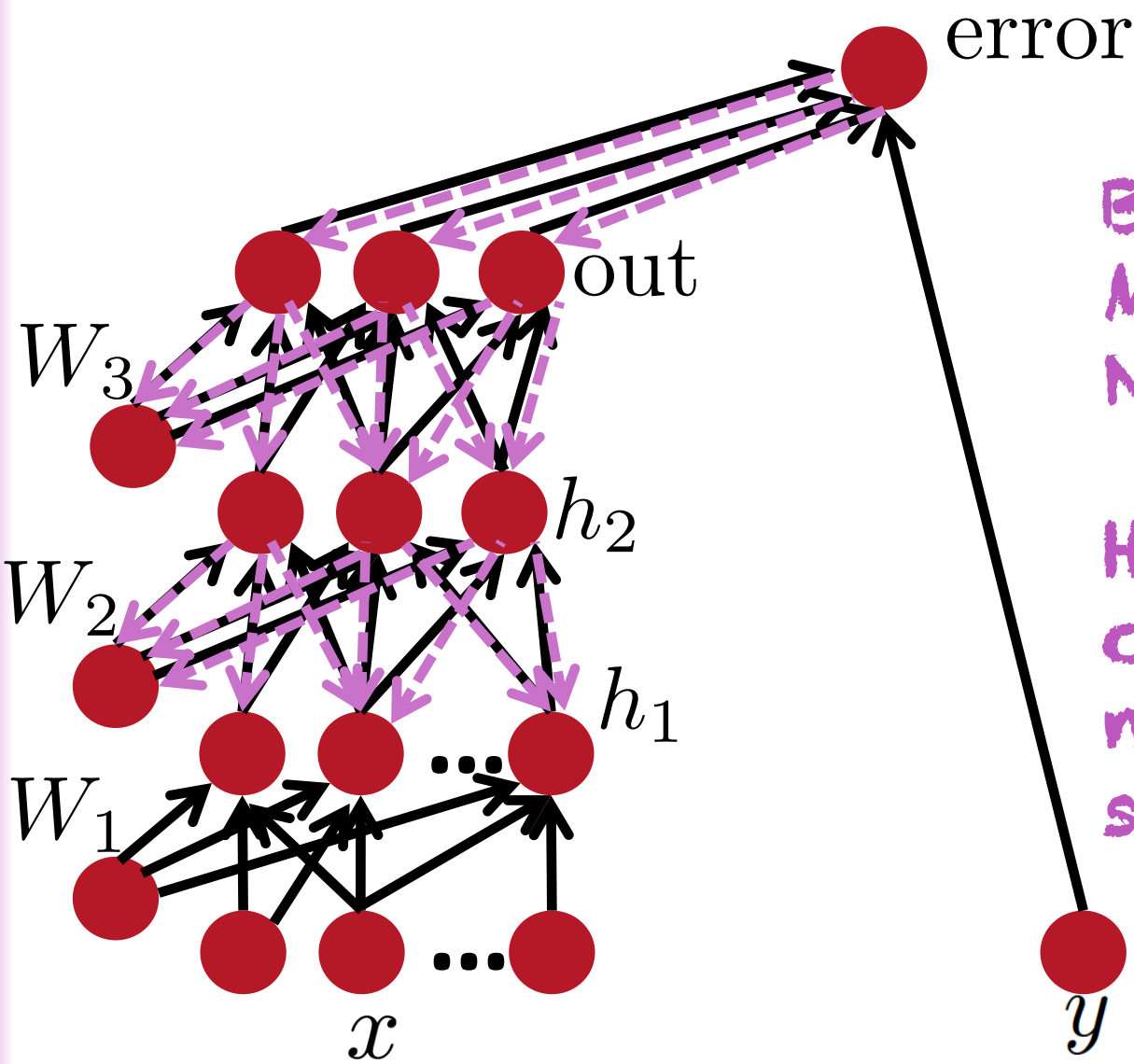
Backprop in Multi-Layer Net:

How outputs could change to make error smaller



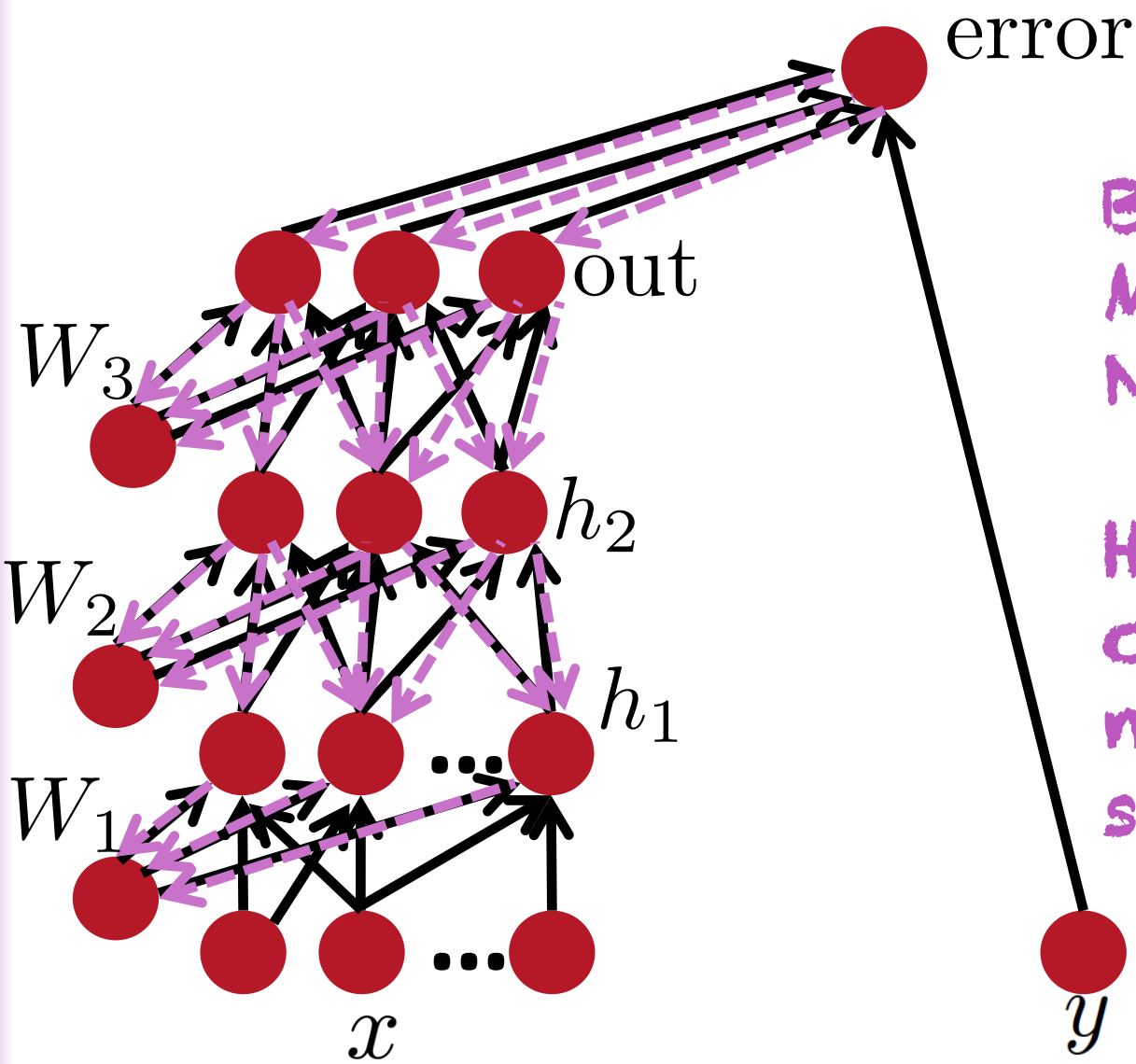
Backprop in Multi-Layer Net:

How h_2 could change to make error smaller



Backprop in Multi-Layer Net:

How h_1 could change to make error smaller

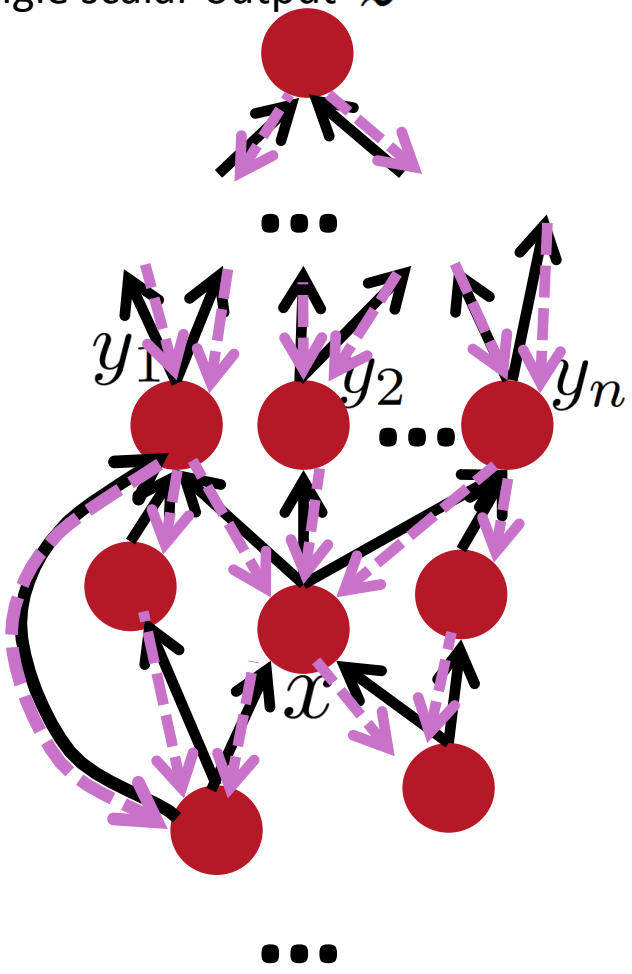


Backprop in Multi-Layer Net:

How W_1 could change to make error smaller

Back-Prop in General Flow Graph

Single scalar output z



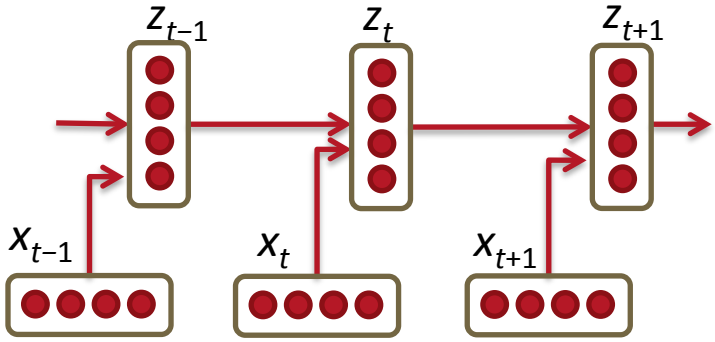
1. Fprop: visit nodes in topo-sort order
   - Compute value of node given predecessors
2. Bprop:
   - initialize output gradient = 1
   - visit nodes in reverse order:
     Compute gradient wrt each node using gradient wrt successors

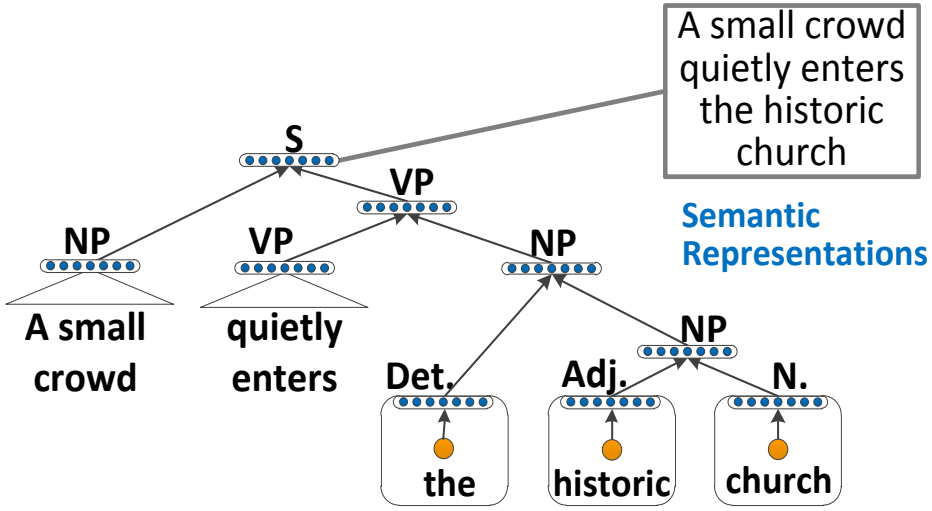
$\{y_1, y_2, \ldots y_n\}$ = successors of x

$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

Back-Prop in Recurrent & Recursive Nets

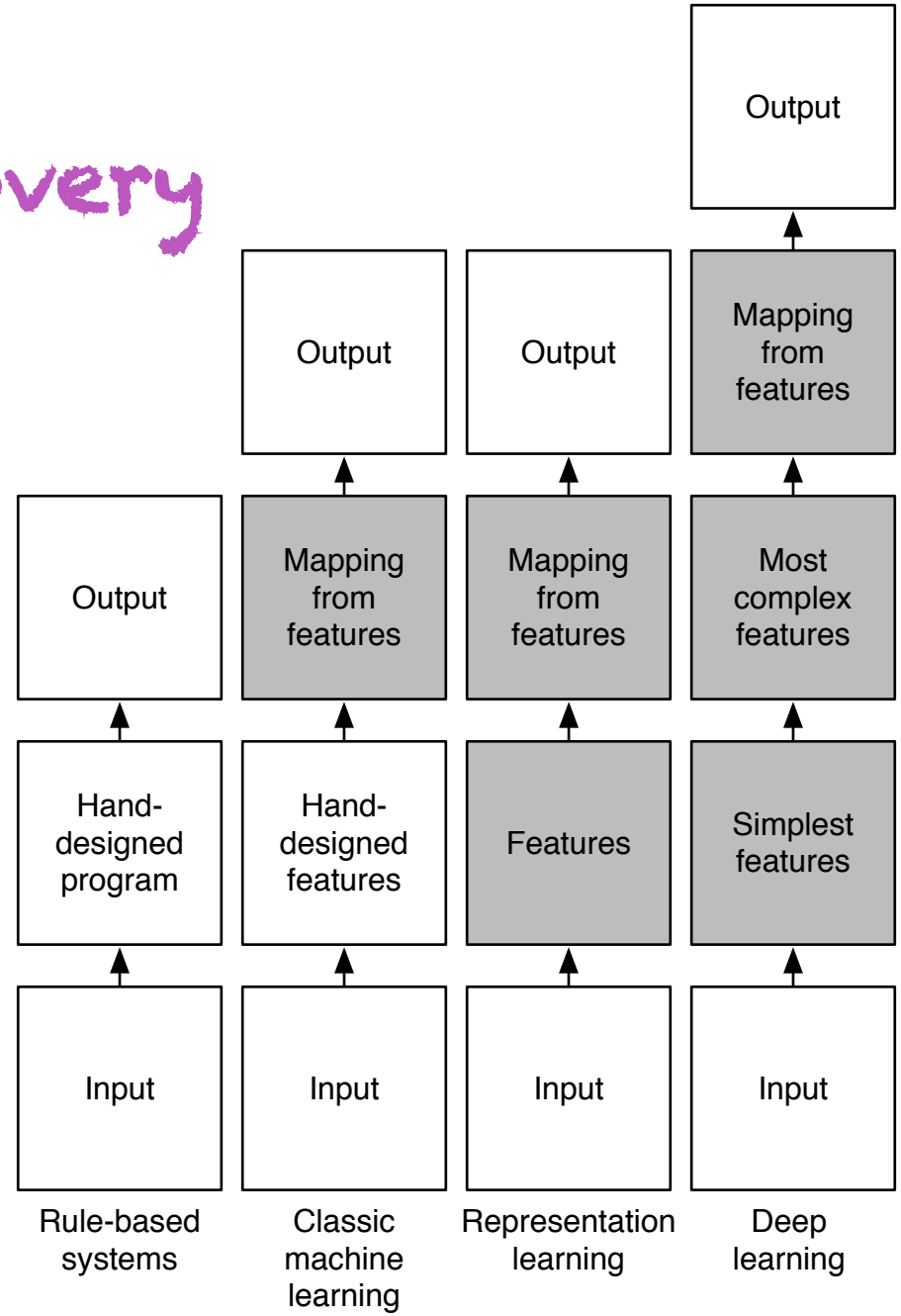
- Replicate a parameterized function over different time steps or nodes of a DAG
- Output state at one time-step / node is used as input for another time-step / node



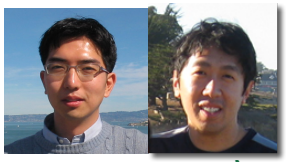


Why is Deep Learning Working so Well?

Automating Feature Discovery

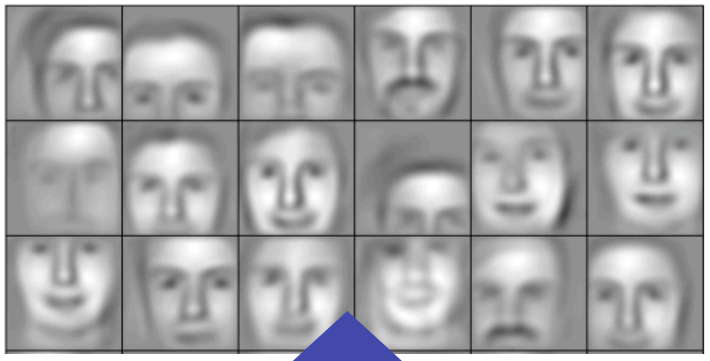


Learning multiple levels of representation



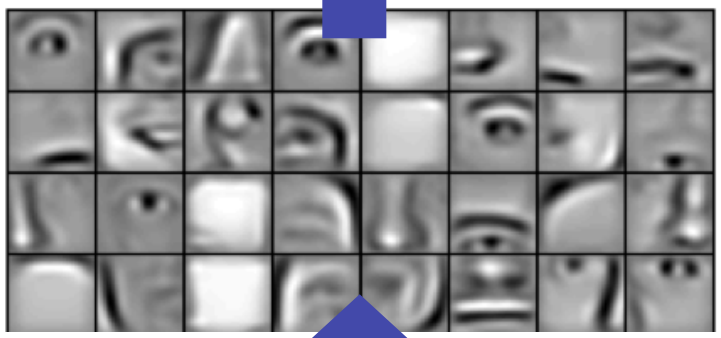
(Lee, Largman, Pham & Ng, NIPS 2009)
(Lee, Grosse, Ranganath & Ng, ICML 2009)

Successive model layers learn deeper intermediate representations



Layer 3

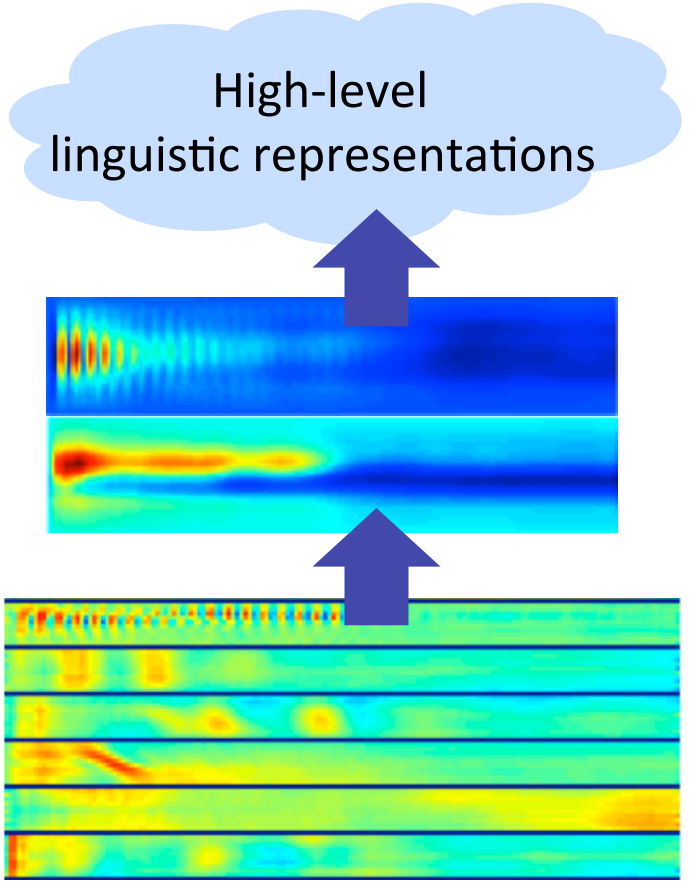
Parts combine to form objects



Layer 2

Layer 1

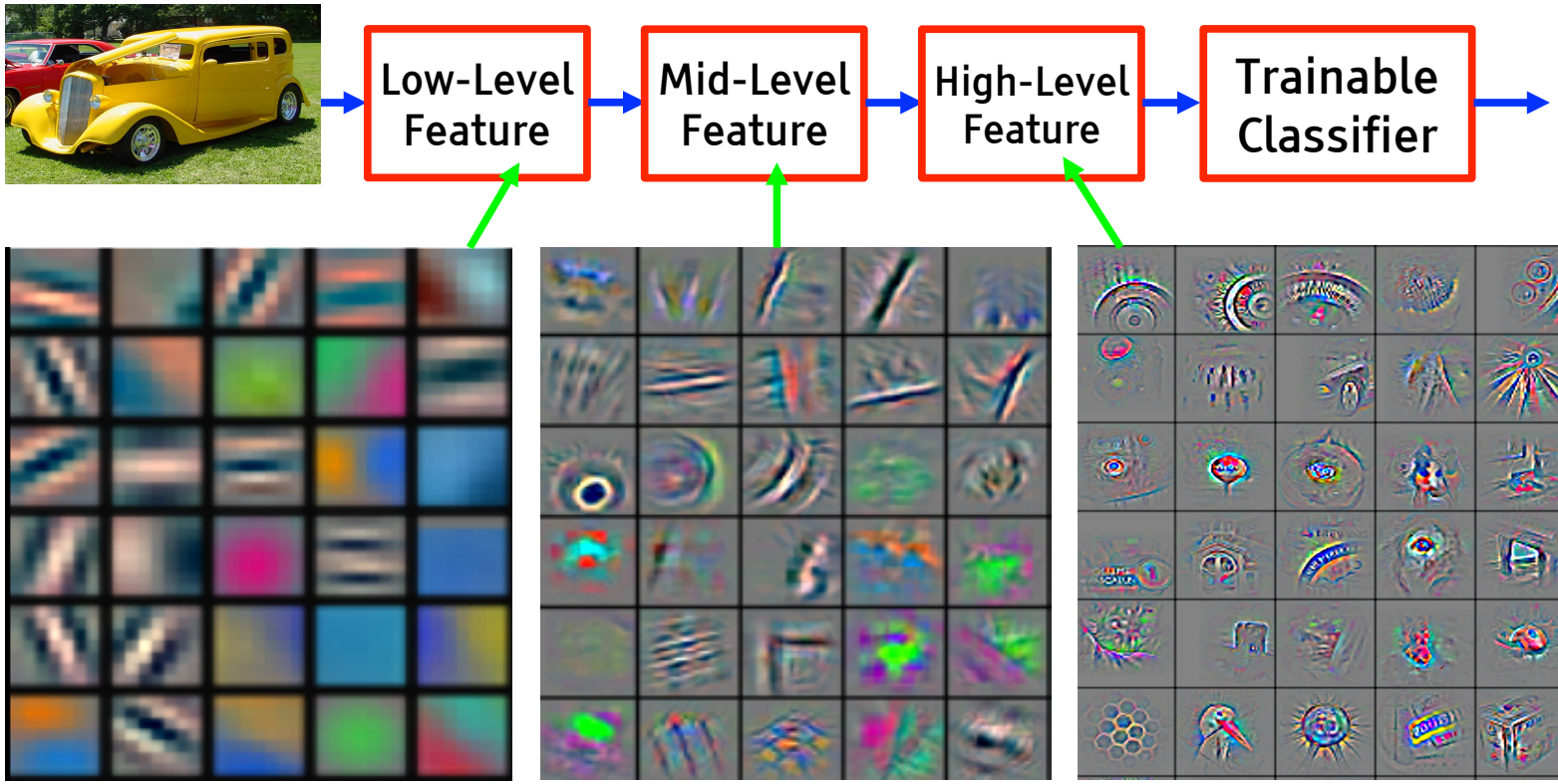
High-level linguistic representations



Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

Why Multiple Layers? The World is Compositional

- Hierarchy of representations with increasing level of abstraction
- Each stage is a kind of trainable feature transform
- Image recognition: Pixel → edge → texton → motif → part → object
- Text: Character → word → word group → clause → sentence → story
- Speech: Sample → spectral band → sound → … → phone → phoneme → word

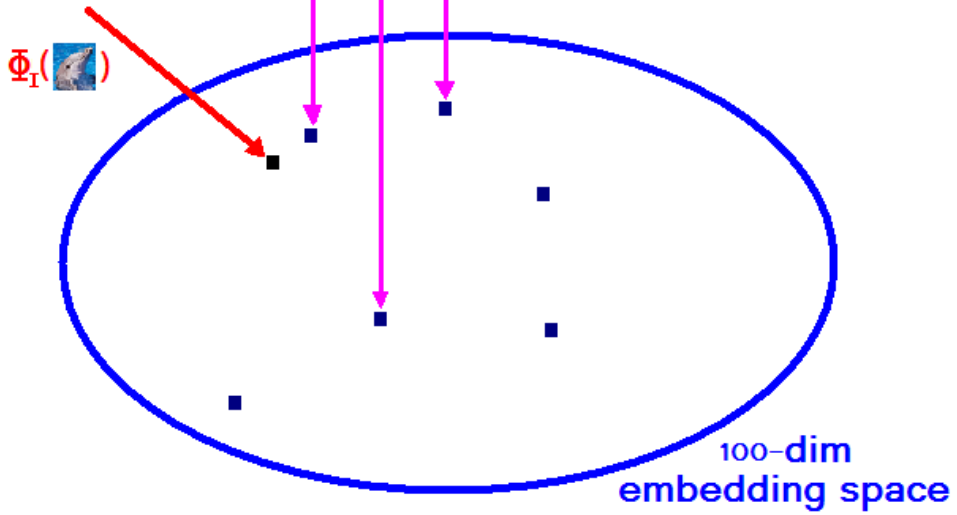


Google Image Search:
Different object types represented in the same space

Google:

S. Bengio, J. Weston & N. Usunier

(IJCAI 2011, NIPS'2010, JMLR 2010, MLJ 2010)



Learn $\Phi_I(\cdot)$ and $\Phi_W(\cdot)$ to optimize precision@k.

Machine Learning, AI & No Free Lunch

- Four key ingredients for ML towards AI

  1. Lots & lots of data
  2. Very flexible models
  3. Enough computing power
  4. Good learning algorithms = powerful priors that can defeat the curse of dimensionality

Bypassing the curse of dimensionality

We need to build compositionality into our ML models

Just as human languages exploit compositionality to give representations and meanings to complex ideas

Exploiting compositionality gives an exponential gain in representational power

Distributed representations / embeddings: feature learning

Deep architecture: multiple levels of feature learning

Prior: compositionality is useful to describe the world around us efficiently

Each feature can be discovered without the need for seeing the exponentially large number of configurations of the other features

- Consider a network whose hidden units discover the following features:
  - Person wears glasses
  - Person is female
  - Person is a child
  - Etc.

If each of *n* feature requires *O(k)* parameters, need *O(nk)* examples

Non-parametric methods would require $O(n^d)$ examples

Exponential advantage of distributed representations

- *Bengio 2009* (Learning Deep Architectures for AI, F & T in ML)
- *Montufar & Morton 2014* (When does a mixture of products contain a product of mixtures? SIAM J. Discr. Math)

- Longer discussion and relations to the notion of priors: ***Deep Learning***, to appear, MIT Press.

- Prop. 2 of ***Pascanu, Montufar & Bengio ICLR'2014***: number of pieces distinguished by 1-hidden-layer rectifier net with *n* units and *d* inputs (i.e. *O(nd)* parameters) is

$$\sum_{j=0}^{d} \binom{n}{j} = O(n^d)$$

The Depth Prior can be Exponentially Advantageous

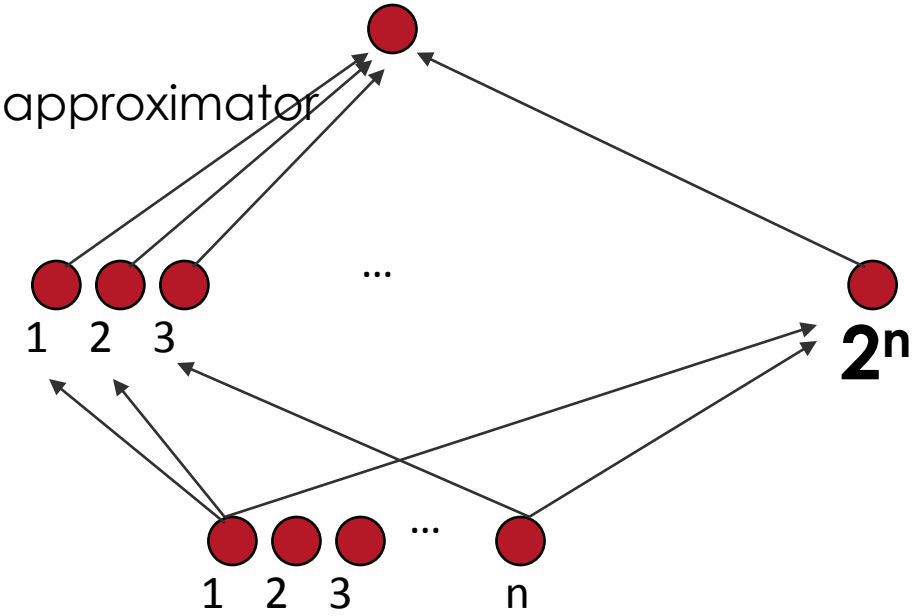
Theoretical arguments:

2 layers of { Logic gates / Formal neurons / RBF units } = universal approximator

RBMs & auto-encoders = universal approximator

Theorems on advantage of depth:
(Hastad et al 86 & 91, Bengio et al 2007, Bengio & Delalleau 2011, Braverman 2011, Pascanu et al 2014, Montufar et al **NIPS 2014**)

Some functions compactly represented with k layers may require exponential size with 2 layers

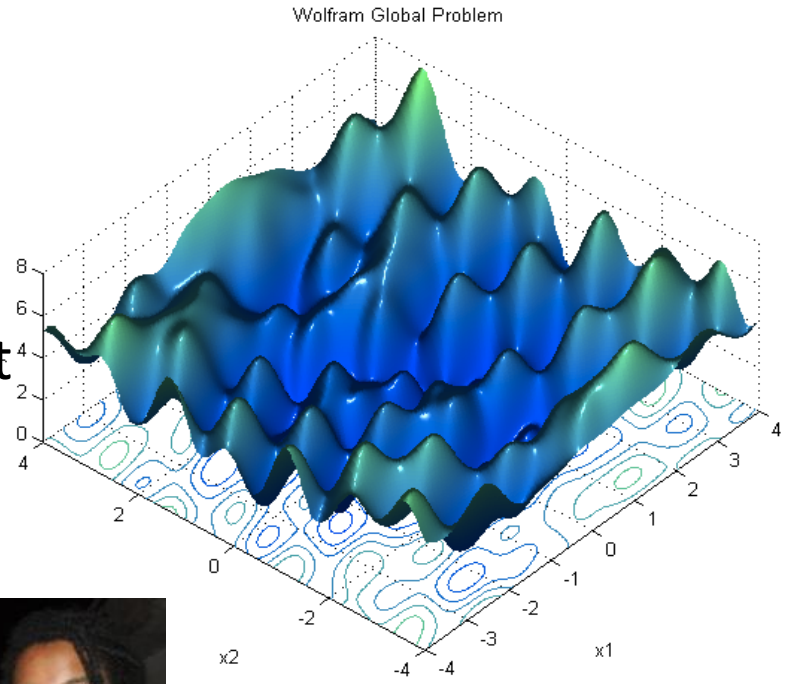


Busting the myth of local minima

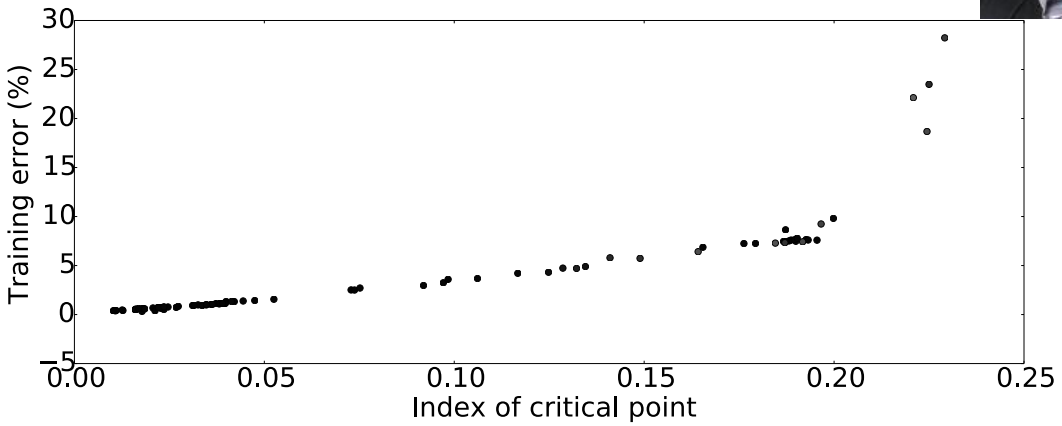
- There are still some researchers who believe that because of the presence of local minima, neural nets should be replaced by kernel machines *(Liu, Lee & Jordan, ICML'2016)*
- Yet, mounting evidence that this is a **myth, for non-tiny nets:**
  - (Pascanu, Dauphin, Ganguli, Bengio, arXiv May 2014): *On the saddle point problem for non-convex optimization*
  - (Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, NIPS' 2014): *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*
  - (Choromanska, Henaff, Mathieu, Ben Arous & LeCun AISTATS'2015): *The Loss Surface of Multilayer Nets*
  - (Daniel Soudry, Yair Carmon, arXiv:1605.08361): *No bad local minima: Data independent training error guarantees for multilayer neural networks*

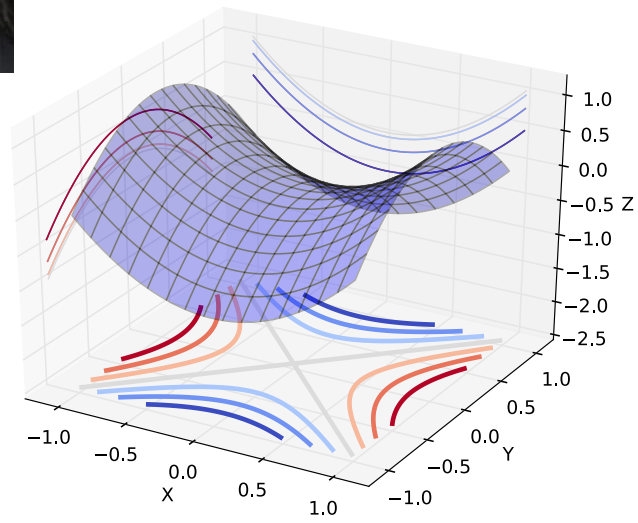
Saddle Points

- Local minima dominate in low-D, but saddle points dominate in high-D
- Most local minima are close to the bottom (global minimum error)



Yann Dauphin





The Supervised Learning Frontier: Attention & Reasoning

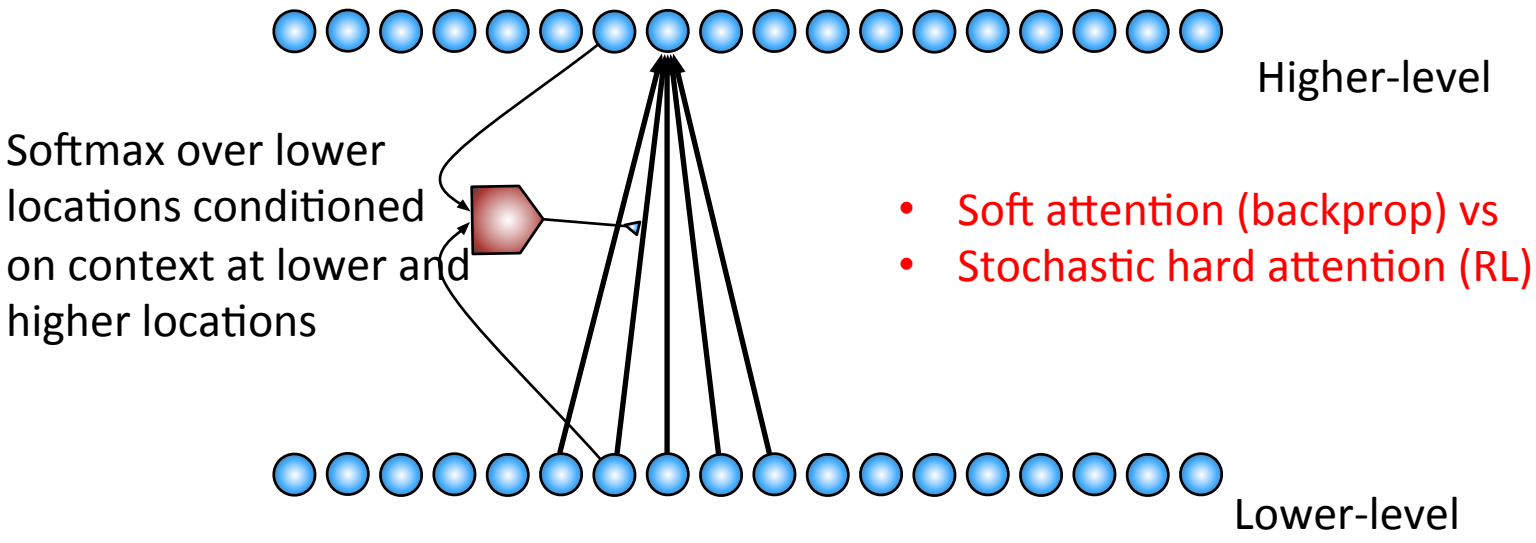
Deep Learning: Beyond Pattern Recognition, towards AI

- Many researchers believed that neural nets could at best be good at pattern recognition
- And they are really good at it!

- But many more ingredients needed towards AI. Recent progress:
  - ATTENTION: focus on a subset of the input elements
  - REASONING: with extensions of recurrent neural networks
    - Memory networks & Neural Turing Machine
  - PLANNING & REINFORCEMENT LEARNING: DeepMind (Atari game playing) & Berkeley (Robotic control)

Attention Mechanism for Deep Learning

- Consider an input (or intermediate) sequence or image
- Consider an upper level representation, which can choose « where to look », by assigning a weight or probability to each input position, as produced by an MLP, applied at each position



- Soft attention (backprop) vs
- Stochastic hard attention (RL)

(Bahdanau, Cho & Bengio, arXiv sept. 2014) following up on *(Graves 2013)* and *(Larochelle & Hinton NIPS 2010)*

End-to-End Machine Translation with Recurrent Nets and Attention Mechanism

(Bahdanau et al 2014, Jean et al 2014, Gulcehre et al 2015, Jean et al 2015)

- Reached the state-of-the-art in one year, from scratch

(a) **English→French (WMT-14)**

	NMT(A)	Google	P-SMT
NMT	32.68	30.6*	**37.03**$^\bullet$
+Cand	33.28	–	
+UNK	33.99	32.7°	
+Ens	**36.71**	**36.9**$^\circ$	

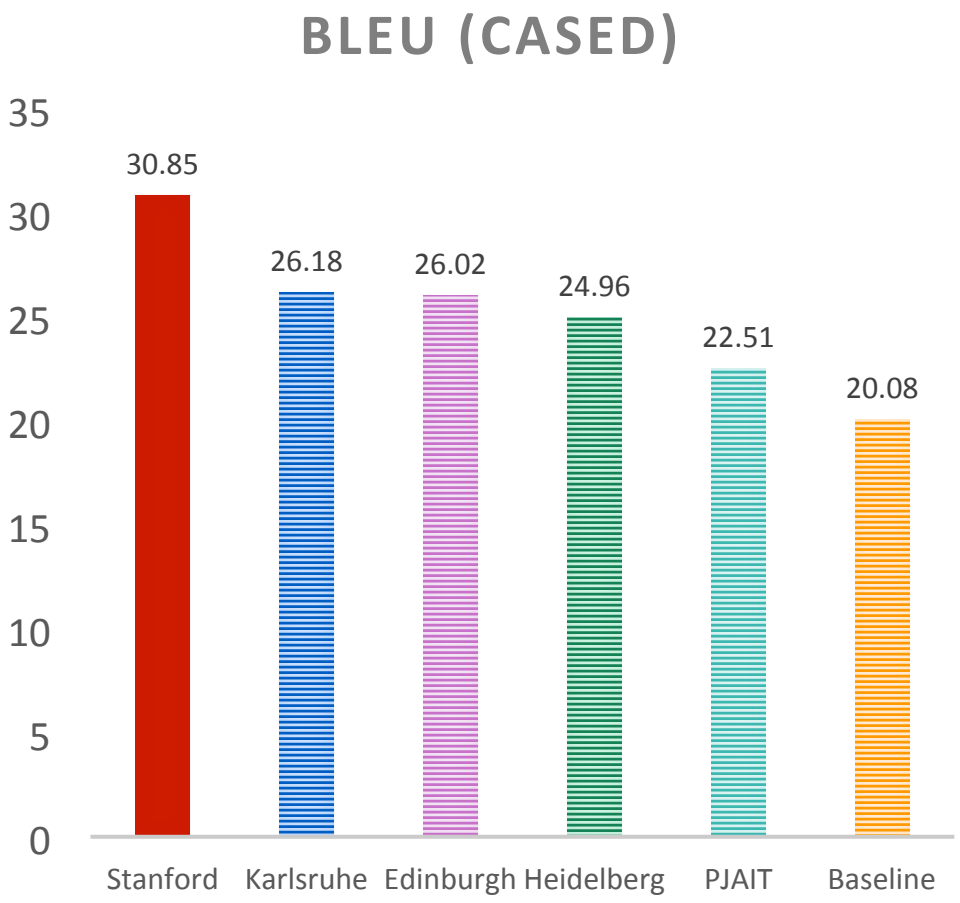
(b) **English→German (WMT-15)**

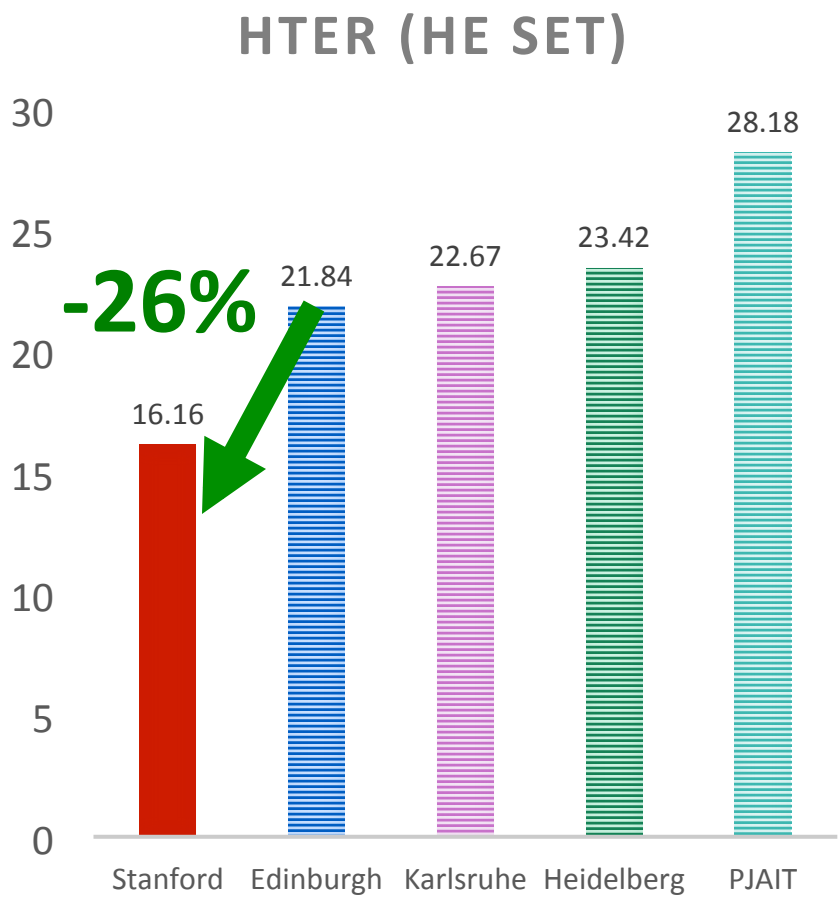
Model	Note
24.8	Neural MT
24.0	U.Edinburgh, Syntactic SMT
23.6	LIMSI/KIT
22.8	U.Edinburgh, Phrase SMT
22.7	KIT, Phrase SMT

(c) **English→Czech (WMT-15)**

Model	Note
18.3	Neural MT
18.2	JHU, SMT+LM+OSM+Sparse
17.6	CU, Phrase SMT
17.4	U.Edinburgh, Phrase SMT
16.1	U.Edinburgh, Syntactic SMT

IWSLT 2015 – Luong & Manning (2015) TED talk MT, English-German



Ongoing progress: combining vision and natural language understanding

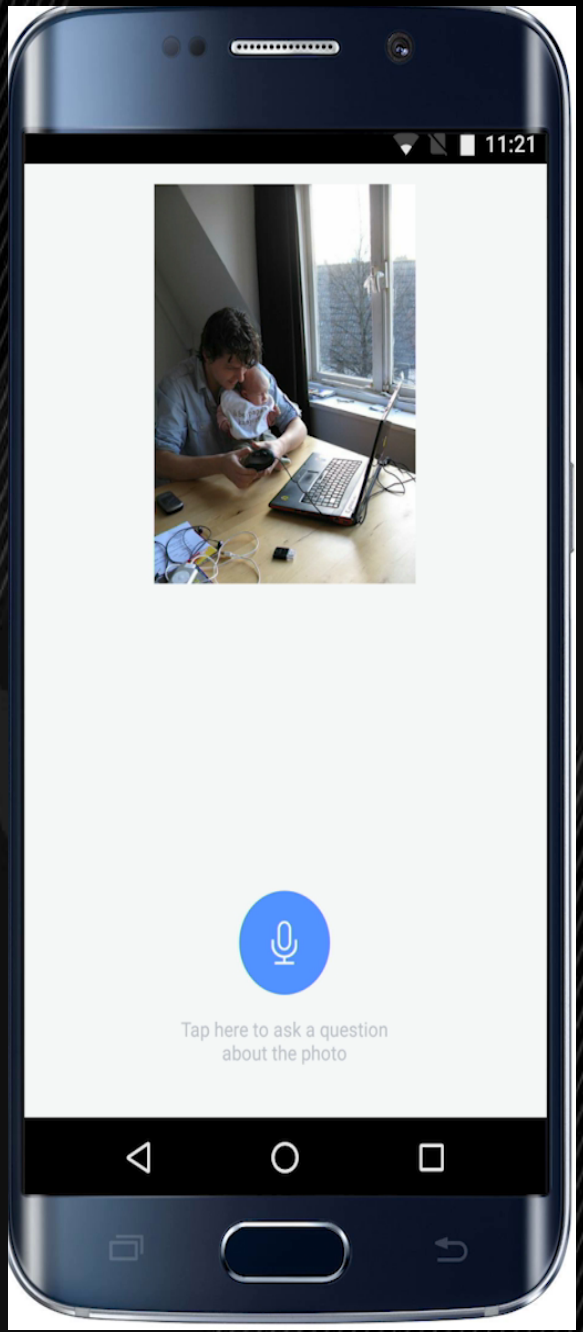
A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor

A stop sign is on a road with a mountain in the background

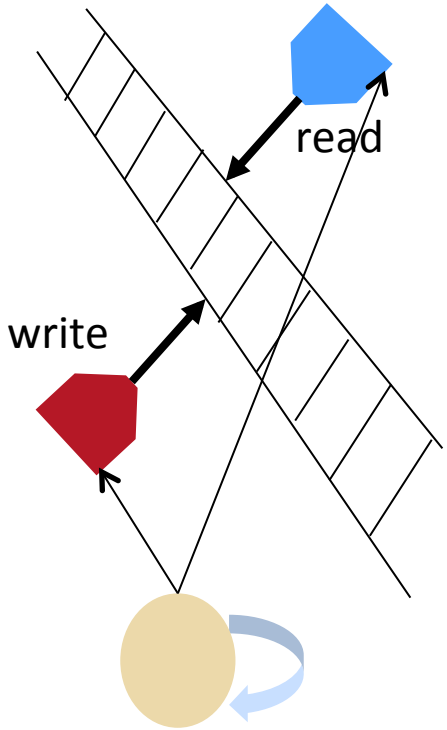
With a lot more data…

visual question answering



Attention Mechanisms for Memory Access Enable Reasoning

- Neural Turing Machines (Graves et al 2014) and Memory Networks (Weston et al 2014)
- Use a form of attention mechanism to control the read and write access into a memory
- The attention mechanism outputs a softmax over memory locations



The next frontier:
to reason and answer questions

Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.

Q: Where is the apple?
A: Bedroom

Brian is a lion.
Julius is a lion.
Julius is white
Bernhard is green

Q: What colour is Brian?
A: White

MILA

Université de Montréal