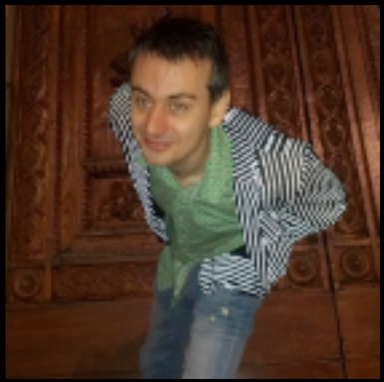
Overview for today

- Natural Language Processing with NNs [~15m]
  - Supervised models
- Unsupervised Learning [~45m]
- Memory in Neural Nets [~30m]

Natural Language Processing

Slides from:




Jason Weston Tomas Mikolov Antoine Bordes Wojciech Zaremba

NLP

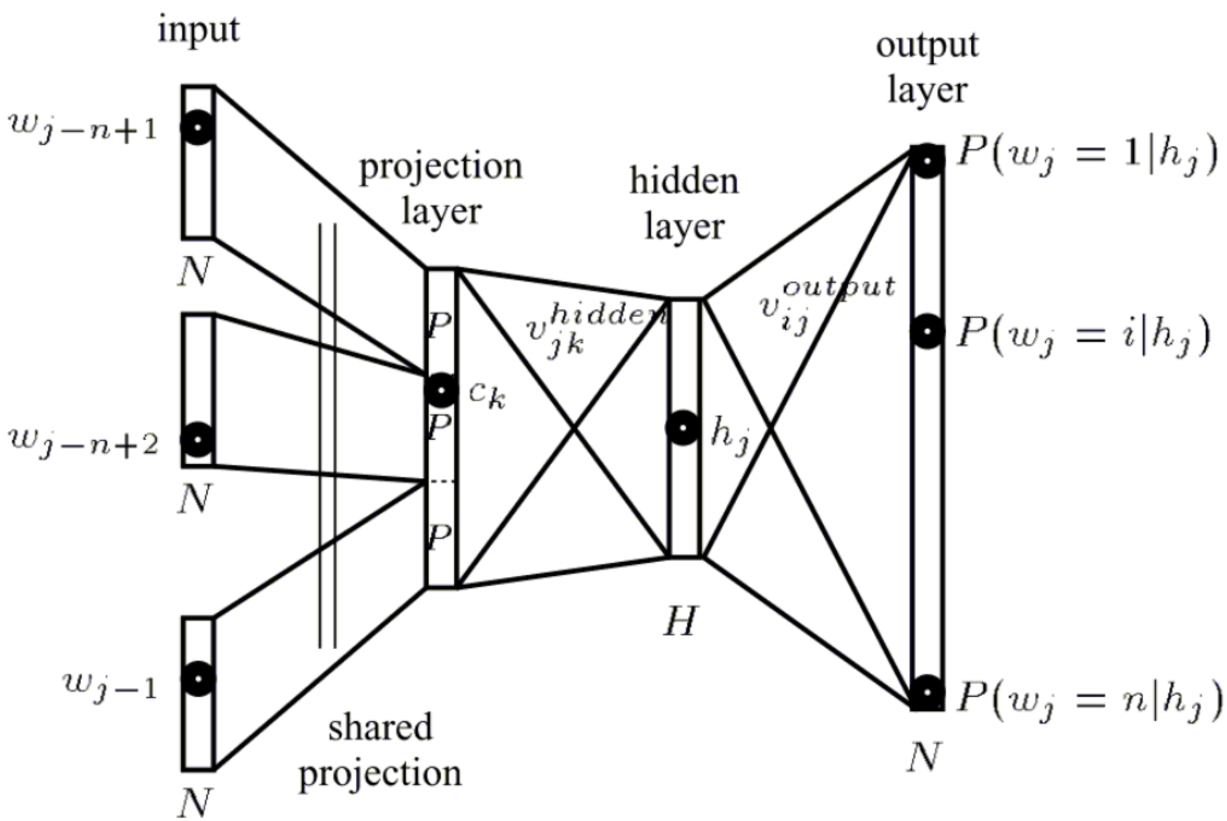
- Many different problems
  - Language modeling
  - Machine translation
  - Q & A
- Recent attempts to address with neural nets
  - Yet to achieve same dramatic gains as vision/speech

Language modeling

- Natural language is a sequence of sequences
- Some sentences are more likely than others:
  - “How are you ?” has a high probability
  - “How banana you ? “ has a low probability

[Slide: Wojciech Zaremba]

Neural Network Language Models



Bengio, Y., Schwenk, H., Sencal, J. S., Morin, F., & Gauvain, J. L. (2006). Neural probabilistic language models. In Innovations in Machine Learning (pp. 137-186). Springer Berlin Heidelberg.

[Slide: Antoine Border & Jason Weston, EMNLP Tutorial 2014]

Recurrent Neural Network Language Models

Key idea: *input to predict next word is current word plus context fed-back from previous word (i.e. remembers the past with recurrent connection).*

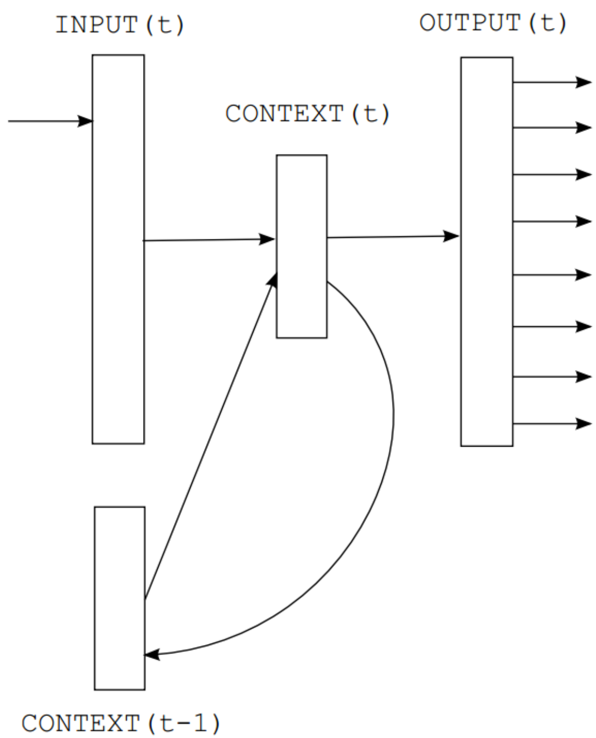
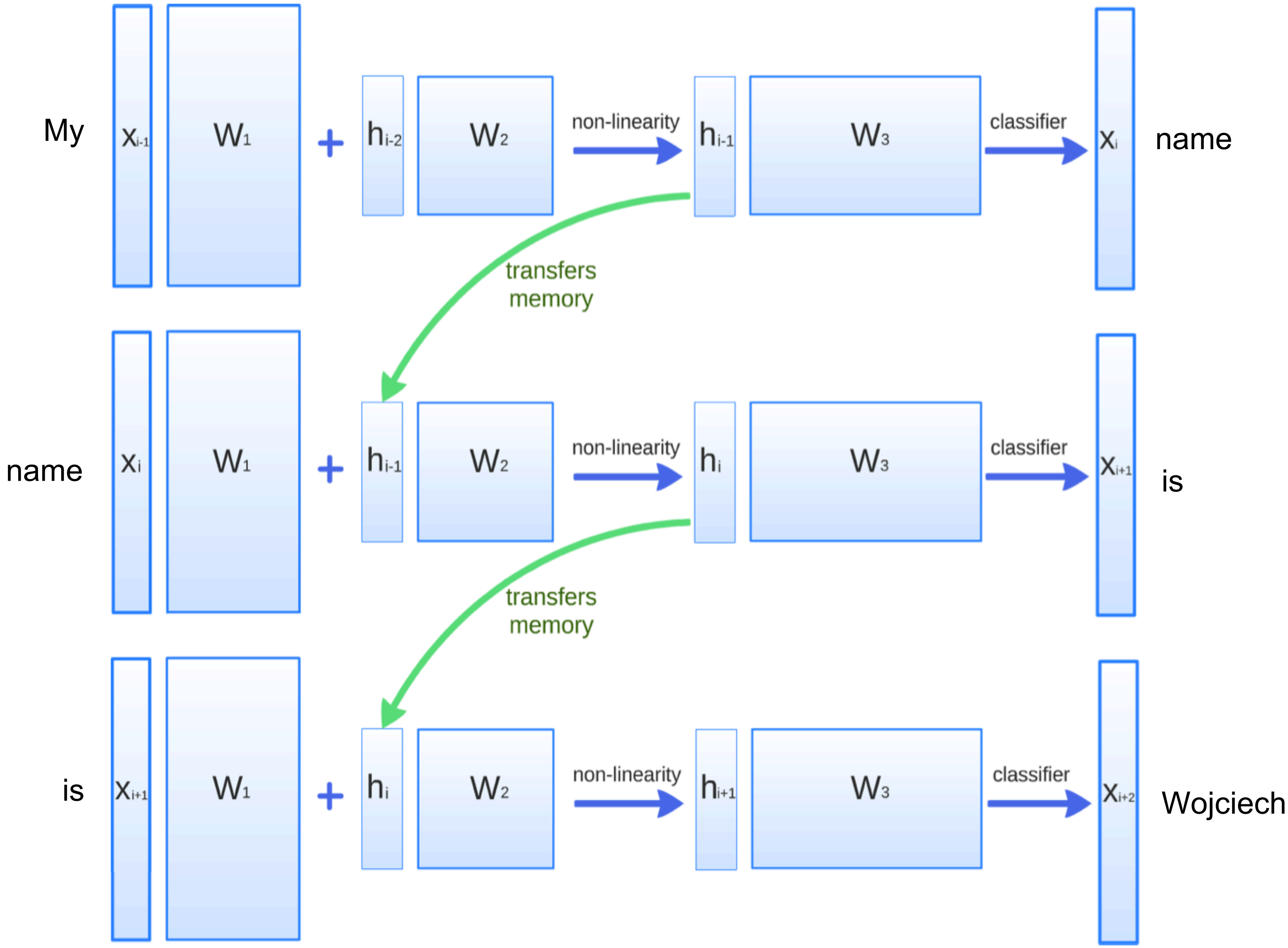


Figure: *Recurrent neural network based LM*

Recurrent neural network based language model. Mikolov et al., Interspeech, '10.

[Slide: Antoine Border & Jason Weston, EMNLP Tutorial 2014]

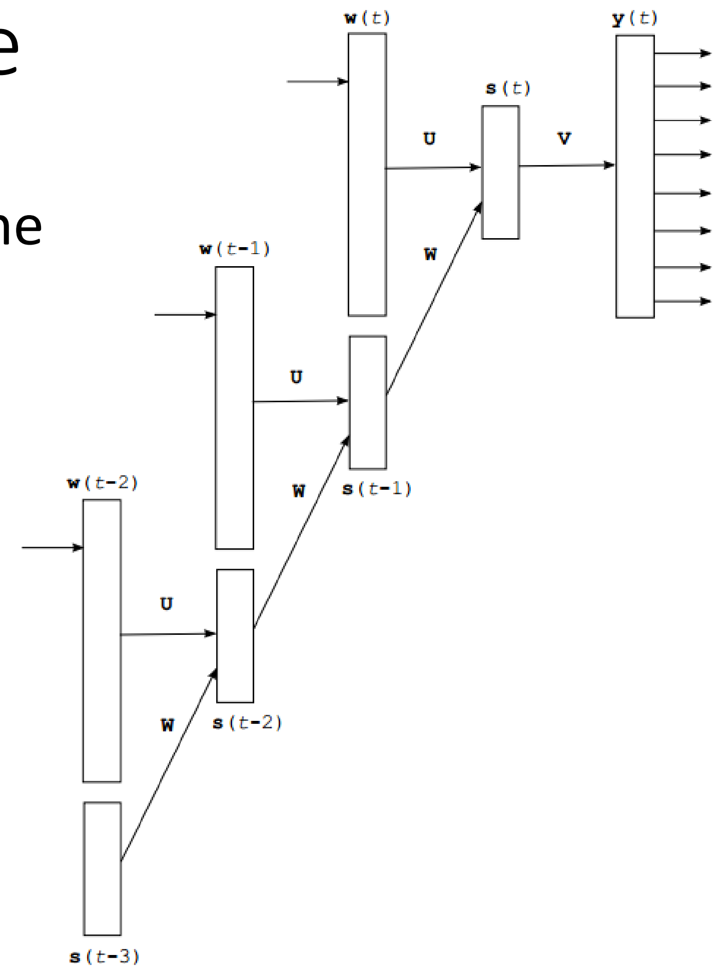
Recurrent neural networks - schema



[Slide: Wojciech Zaremba]

Backpropagation through time

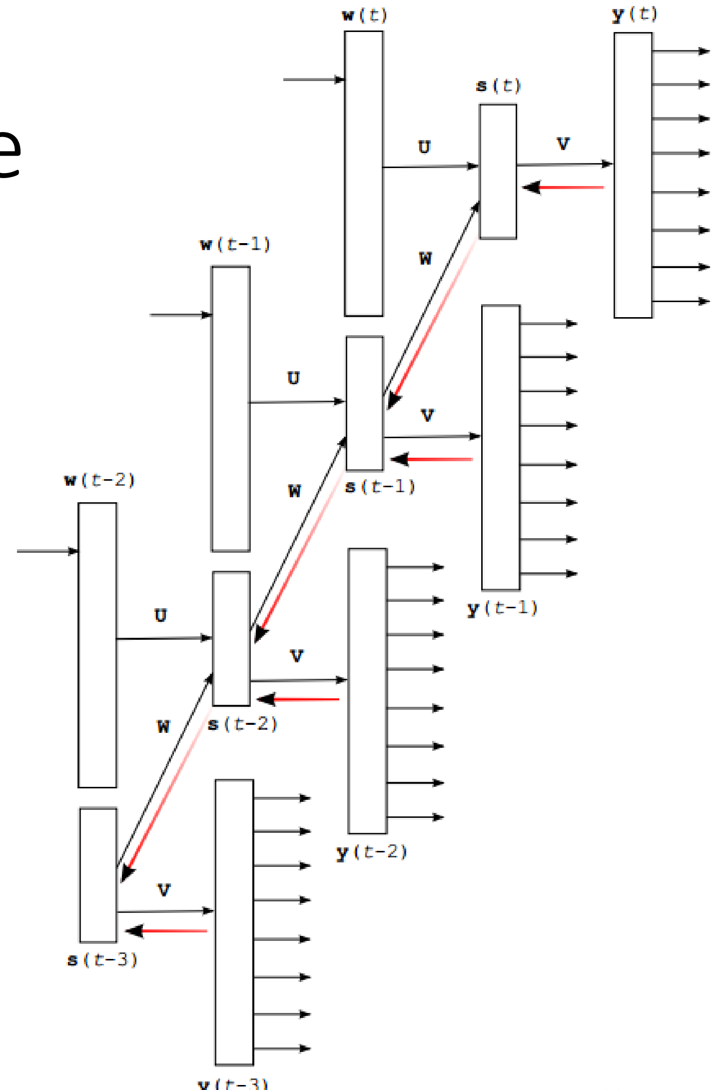
- The intuition is that we unfold the RNN in time
- We obtain deep neural network with shared weights **U** and **W**



[Slide: Thomas Mikolov, COLING 2014]

Backpropagation through time

- We train the unfolded RNN using normal backpropagation + SGD
- In practice, we limit the number of unfolding steps to 5 – 10
- It is computationally more efficient to propagate gradients after few training examples (batch mode)



[Slide: Thomas Mikolov, COLING 2014]

NNLMS vs. RNNS: Penn Treebank Results (Mikolov)

Model	Weight	PPL
3-gram with Good-Turing smoothing (GT3)	0	165.2
5-gram with Kneser-Ney smoothing (KN5)	0	141.2
5-gram with Kneser-Ney smoothing + cache	0.0792	125.7
Maximum entropy model	0	142.1
Random clusterings LM	0	170.1
Random forest LM	0.1057	131.9
Structured LM	0.0196	146.1
Within and across sentence boundary LM	0.0838	116.6
Log-bilinear LM	0	144.5
Feedforward NNLM	0	140.2
Syntactical NNLM	0.0828	131.3
Combination of static RNNLMs	0.3231	102.1
Combination of adaptive RNNLMs	0.3058	101.0
ALL	1	**83.5**

Recent uses of NNLMs and RNNs to improve machine translation:

Fast and Robust NN Joint Models for Machine Translation, Devlin et al, ACL '14.

Also Kalchbrenner '13, Sutskever et al., '14., Cho et al., '14. .

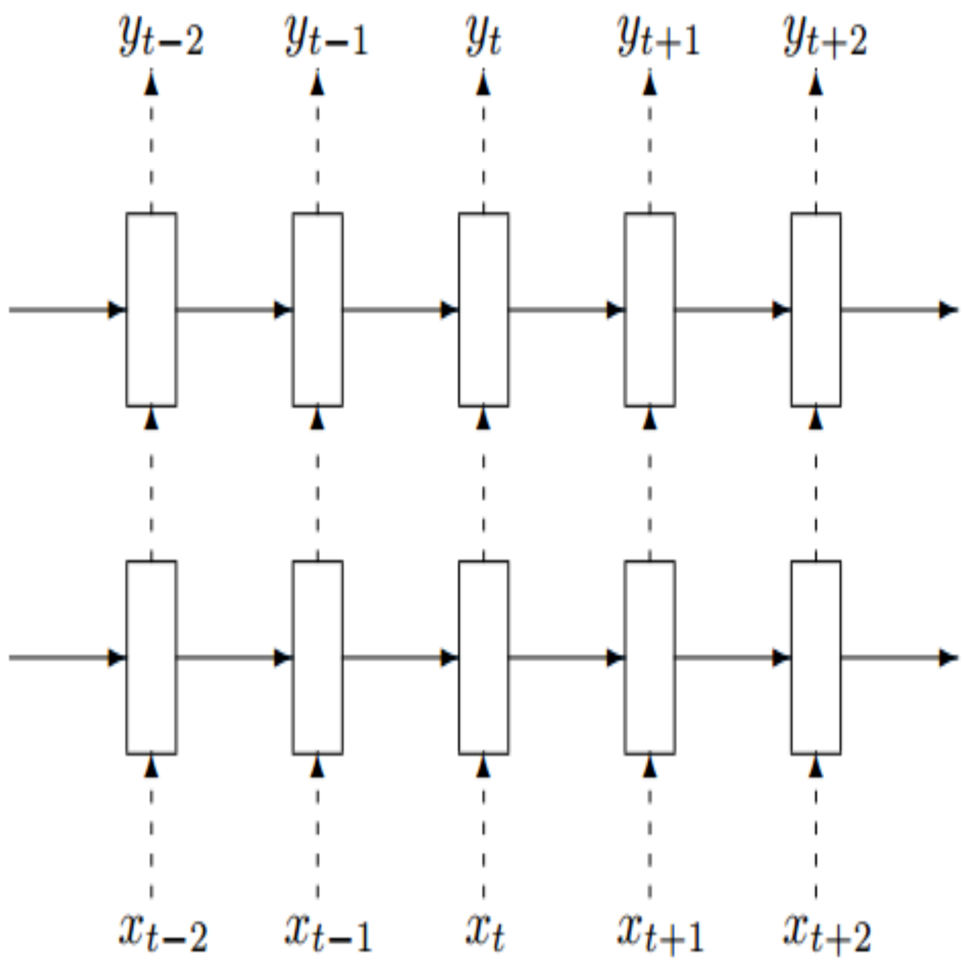
[Slide: Antoine Border & Jason Weston, EMNLP Tutorial 2014]

Language modelling – RNN samples

the meaning of life is that only if an end would be of the whole supplier. widespread rules are regarded as the companies of refuses to deliver. in balance of the nation's information and loan growth associated with the carrier thrifts are in the process of slowing the seed and commercial paper.

[Slide: Wojciech Zaremba]

More depth gives more power

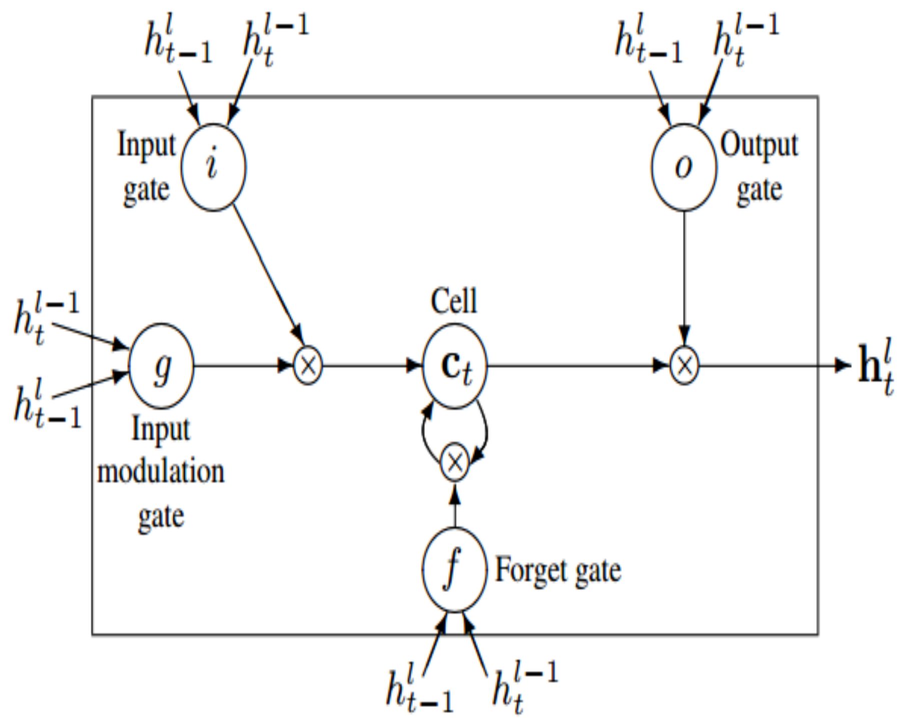


[Slide: Wojciech Zaremba]

LSTM - Long Short Term Memory

[Hochreiter and Schmidhuber, Neural Computation 1997]

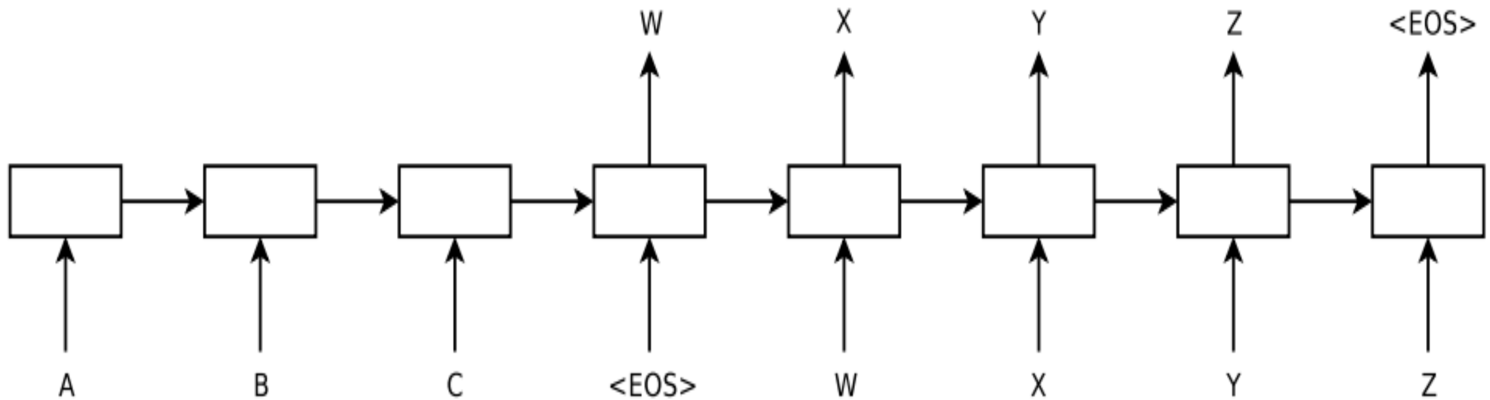
- Ad-hoc way of modelling long dependencies
- Many alternative ways of modelling it
- Next hidden state is modification of previous hidden state (so information doesn't decay too fast).



For simple explanation, see [Recurrent Neural Network Regularization, Wojciech Zaremba, Ilya Sutskever, Oriol Vinyals, arXiv 1409.2329, 2014]

[Slide: Wojciech Zaremba]

RNN-LSTMs for Machine Translation



[Sutskever et. al. (2014)]

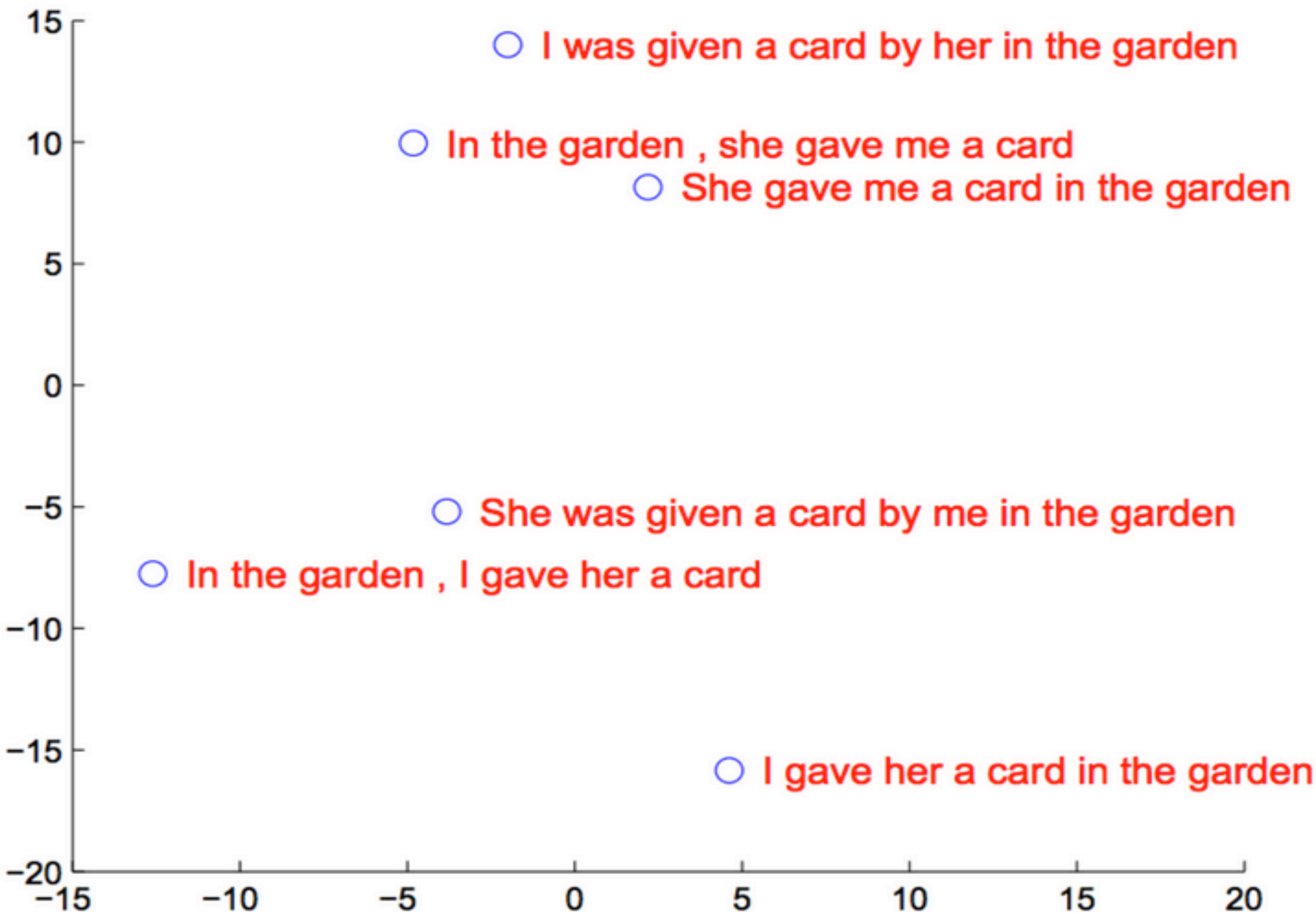
Sequence to Sequence Learning with Neural Networks,
Ilya Sutskever, Oriol Vinyals, Quoc Le, NIPS 2014

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, EMNLP 2014

[Slide: Wojciech Zaremba]

Visualizing Internal Representation

t-SNE projection of network state at end of input sentence



Sequence to Sequence Learning with Neural Networks,
Ilya Sutskever, Oriol Vinyals, Quoc Le, NIPS 2014

Translation - examples

- FR: Les avionneurs se querellent au sujet de la largeur des sièges alors que de grosses commandes sont en jeu

- Google Translate: Aircraft manufacturers are quarreling about the seat width as large orders are at stake

- LSTM: Aircraft manufacturers are concerned about the width of seats while large orders are at stake

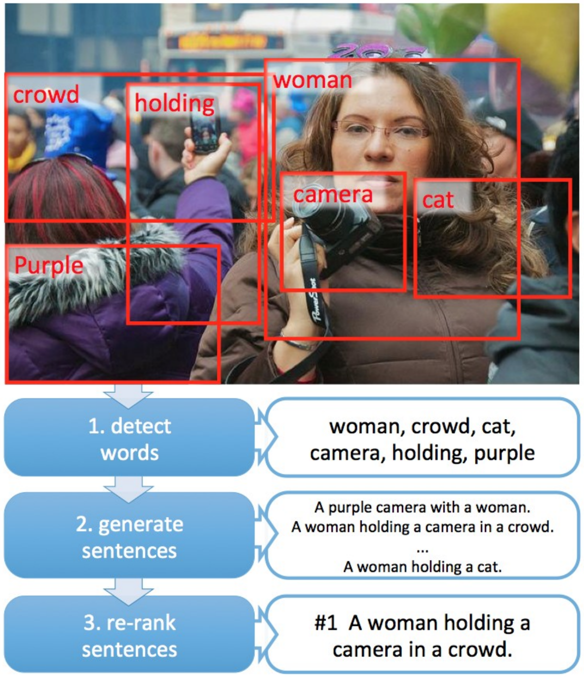
- Ground Truth: Jet makers feud over seat width with big orders at stake

[Sequence to Sequence Learning with Neural Networks,
Ilya Sutskever, Oriol Vinyals, Quoc Le, NIPS 2014]

[Slide: Wojciech Zaremba]

Image Captioning: Vision + NLP

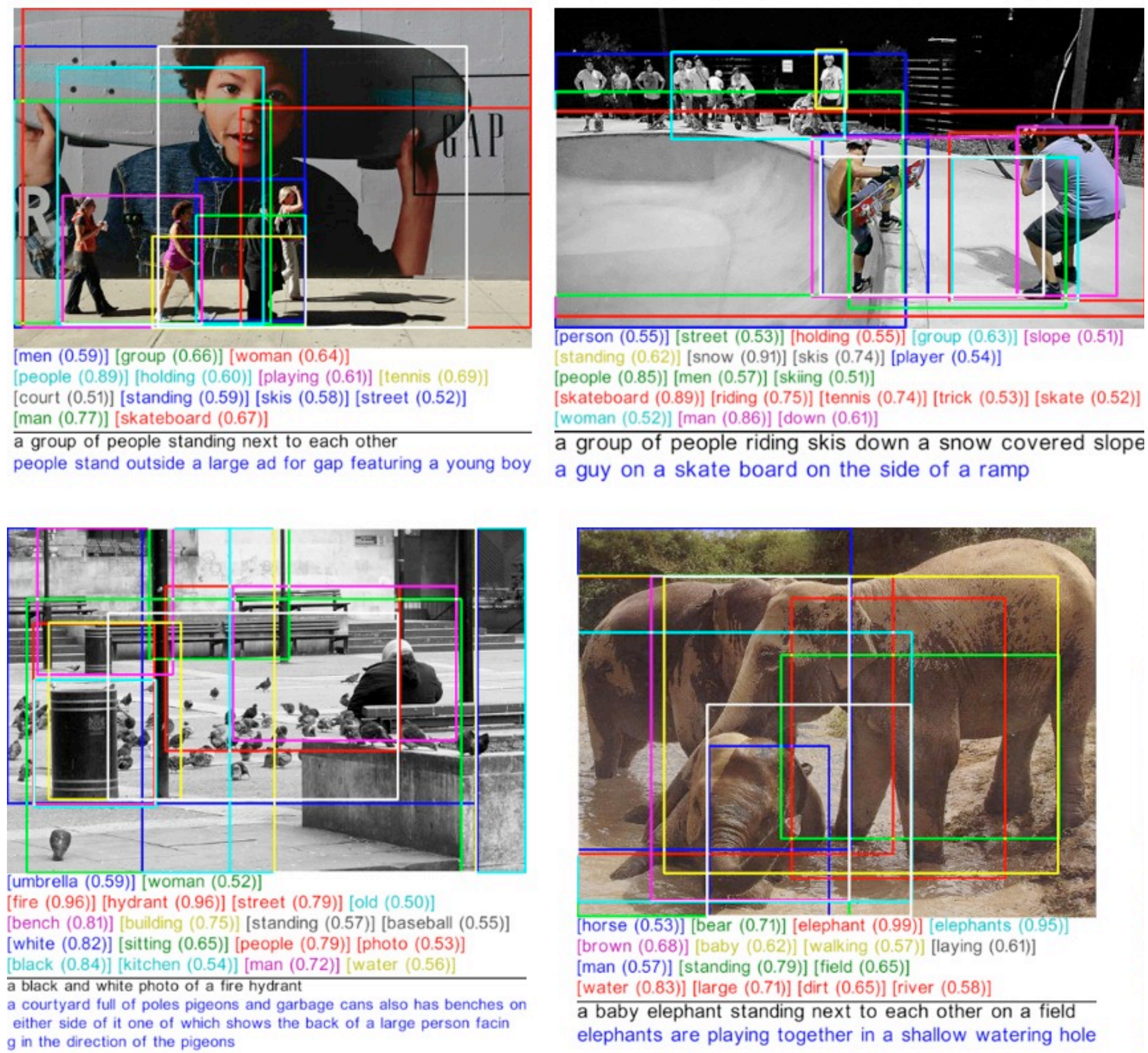
- Generate short text descriptions of image, given just picture.
- Use Convnet to extract image features
- RNN or LSTM model takes image features as input, generates text



Many recent works on this:

- Baidu/UCLA: Explain Images with Multimodal Recurrent Neural Networks
- Toronto: Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models
- Berkeley: Long-term Recurrent Convolutional Networks for Visual Recognition and Description
- Google: Show and Tell: A Neural Image Caption Generator
- Stanford: Deep Visual-Semantic Alignments for Generating Image Description
- UML/UT: Translating Videos to Natural Language Using Deep Recurrent Neural Networks
- Microsoft/CMU: Learning a Recurrent Visual Representation for Image Caption Generation
- Microsoft: From Captions to Visual Concepts and Back

Image Captioning Examples

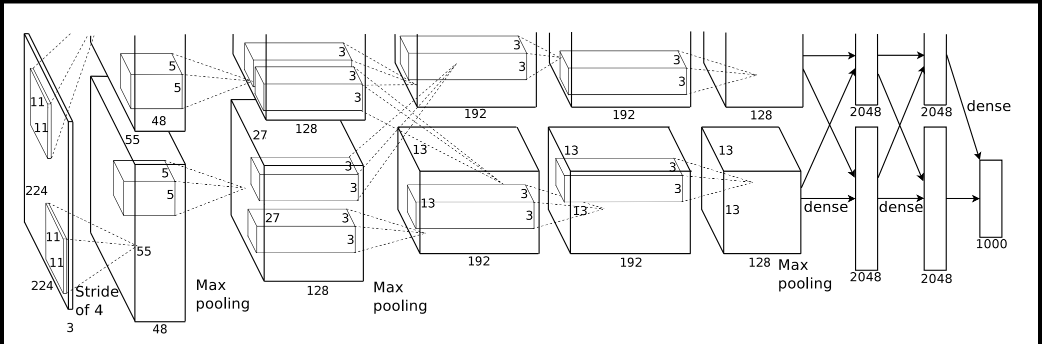


From Captions to Visual Concepts and Back, Hao Fang∗ Saurabh Gupta∗ Forrest Iandola∗ Rupesh K. Srivastava∗, Li Deng Piotr Dollar, Jianfeng Gao Xiaodong He, Margaret Mitchell John C. Platt, C. Lawrence Zitnick, Geoffrey Zweig, CVPR 2015.

Unsupervised Learning

Motivation

- Most successes obtained with supervised models, e.g. Convnets



- Unsupervised learning methods less successful

- But likely to be very important in long-term

Historical Note

- Deep Learning revival started in ~2006
  - Hinton & Salakhudinov Science paper on RBMs
- Unsupervised Learning was focus from 2006-2012
- In ~2012 great results in vision, speech with supervised methods appeared
  - Less interest in unsupervised learning

Arguments for Unsupervised Learning

- Want to be able to exploit unlabeled data
  - Vast amount of it often available
  - Essentially free
- Good regularizer for supervised learning
  - Helps generalization
  - Transfer learning
  - Zero / one-shot learning

Another Argument for Unsupervised Learning

When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information.

You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds.

So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information: from the input itself.

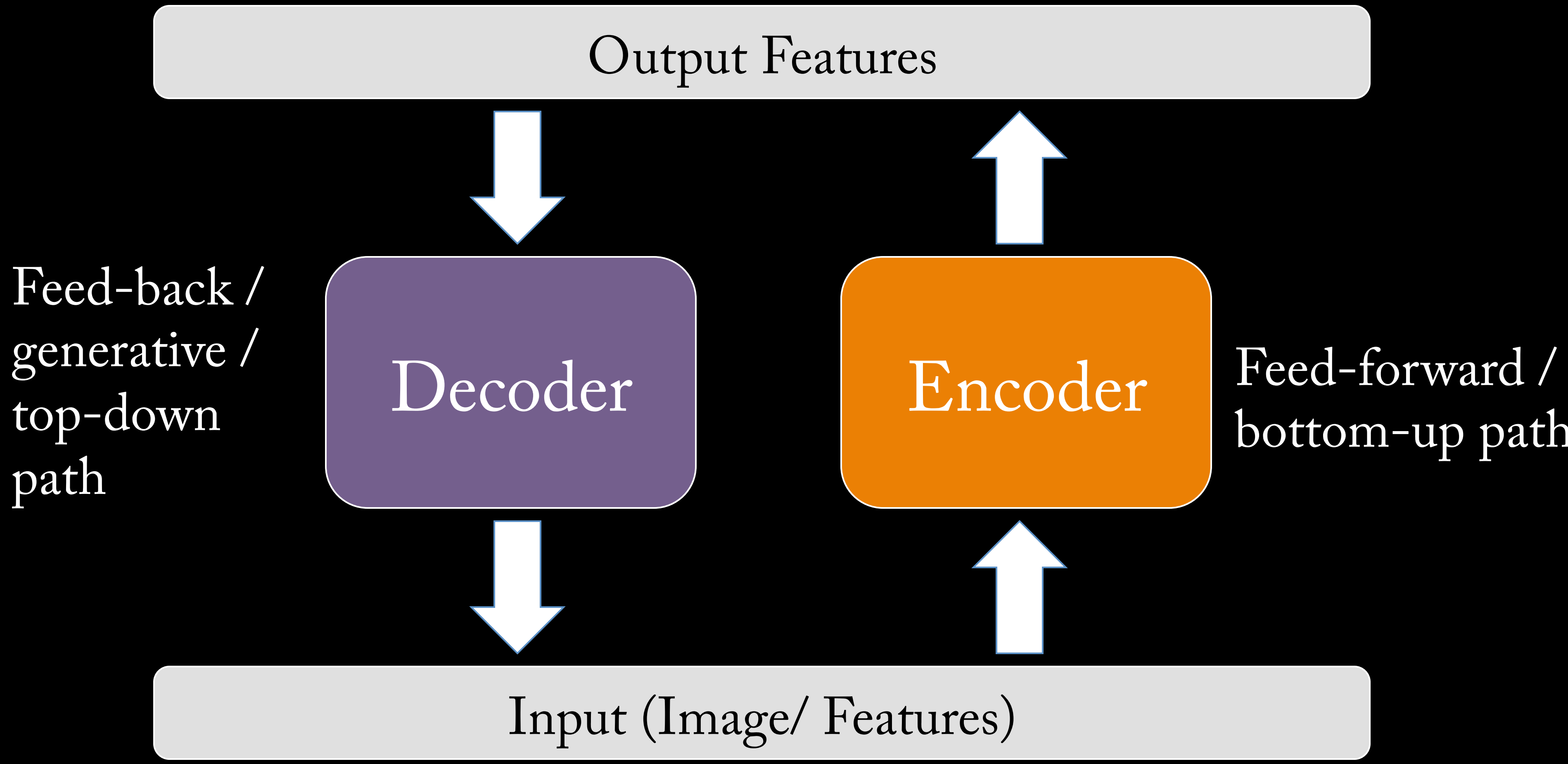
— Geoffrey Hinton, 1996

Taxonomy of Approaches

- Autoencoder (most unsupervised Deep Learning methods)
  - RBMs / DBMs
  - Denoising autoencoders
  - Predictive sparse decomposition
- Decoder-only
  - Sparse coding
  - Deconvolutional Nets
- Encoder-only
  - Implicit supervision, e.g. from video
- Adversarial Networks

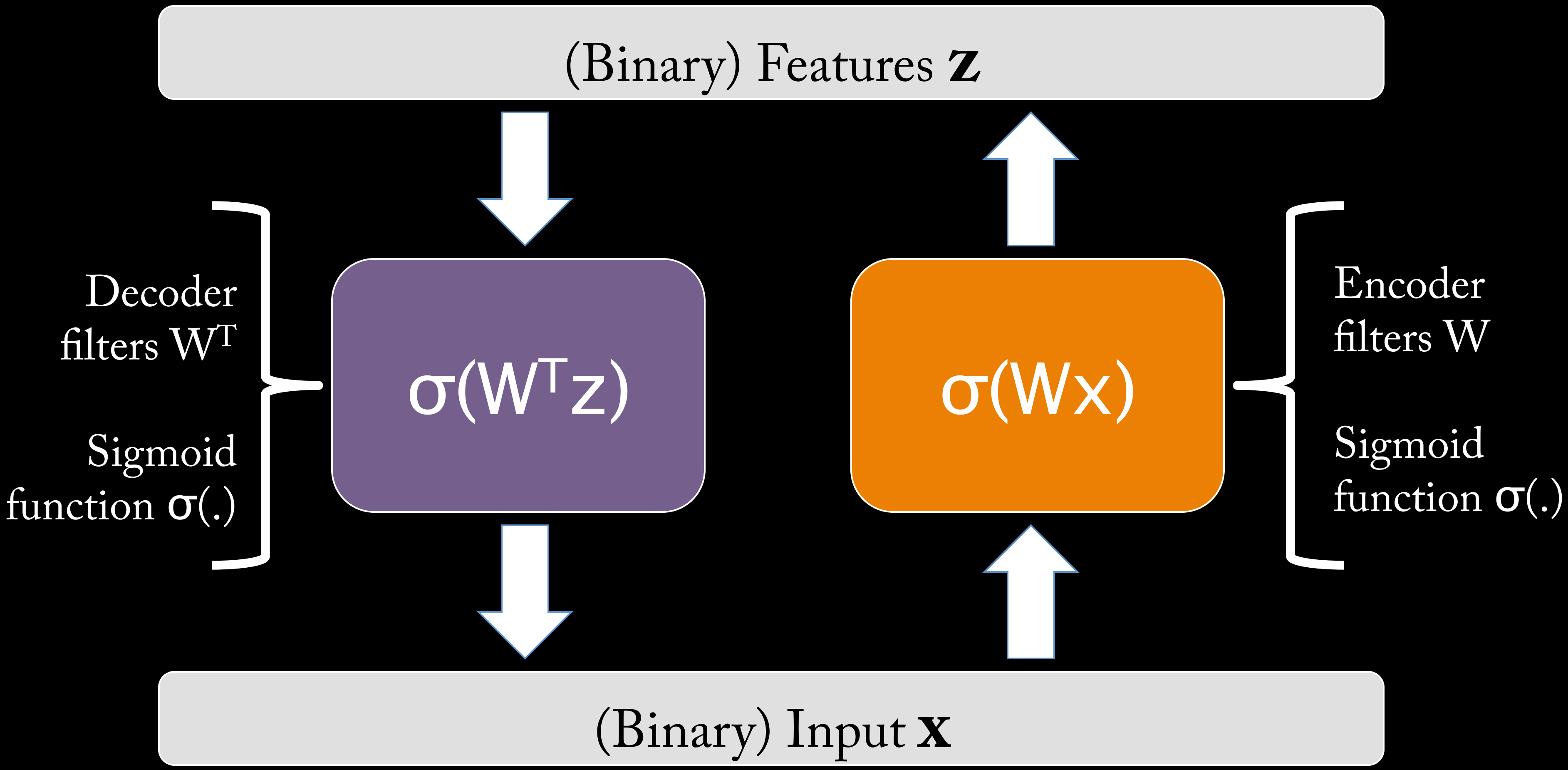
Loss involves some kind of reconstruction error

Auto-Encoder



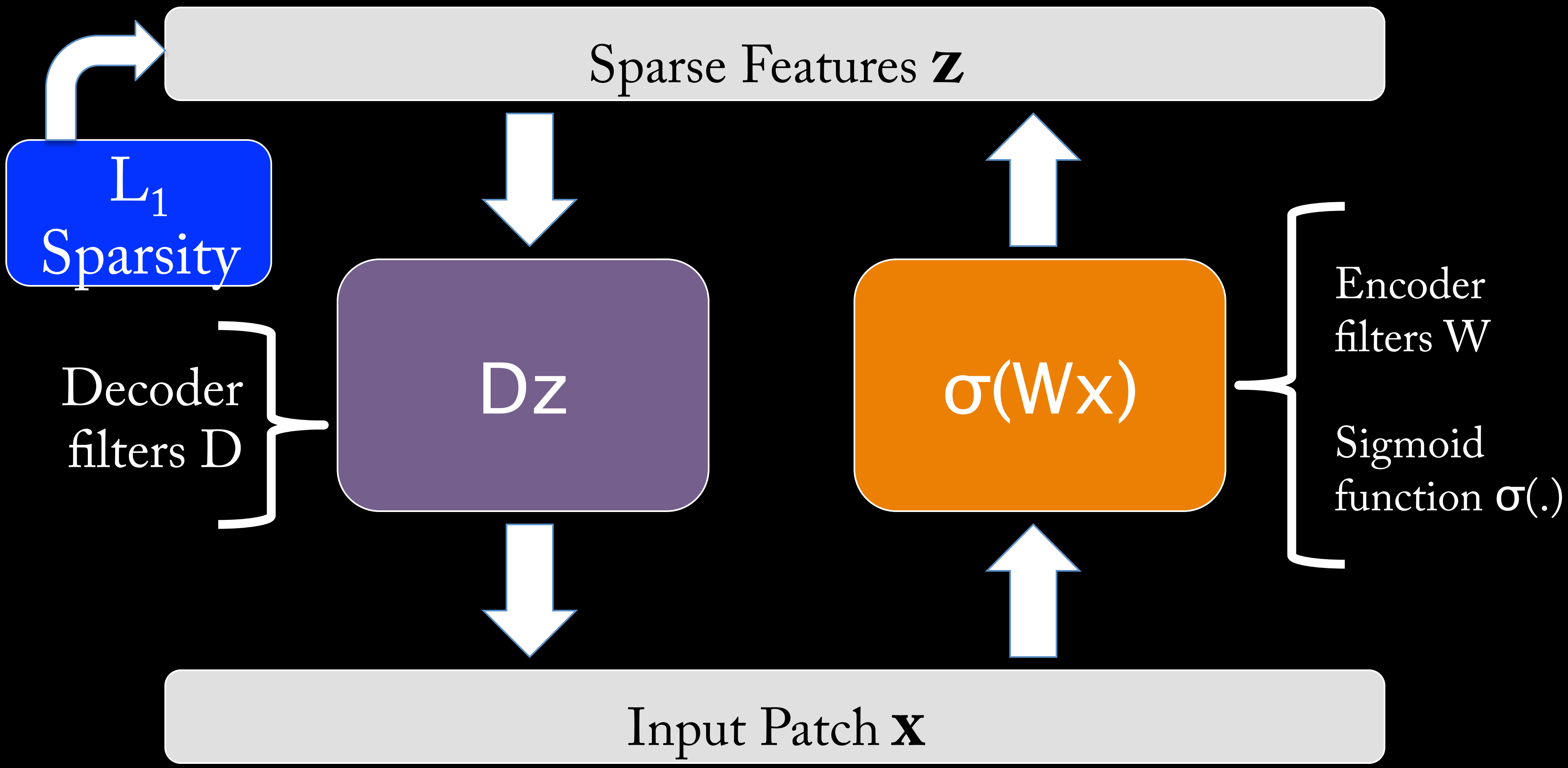
Auto-Encoder Example 1

- Restricted Boltzmann Machine [Hinton '02]



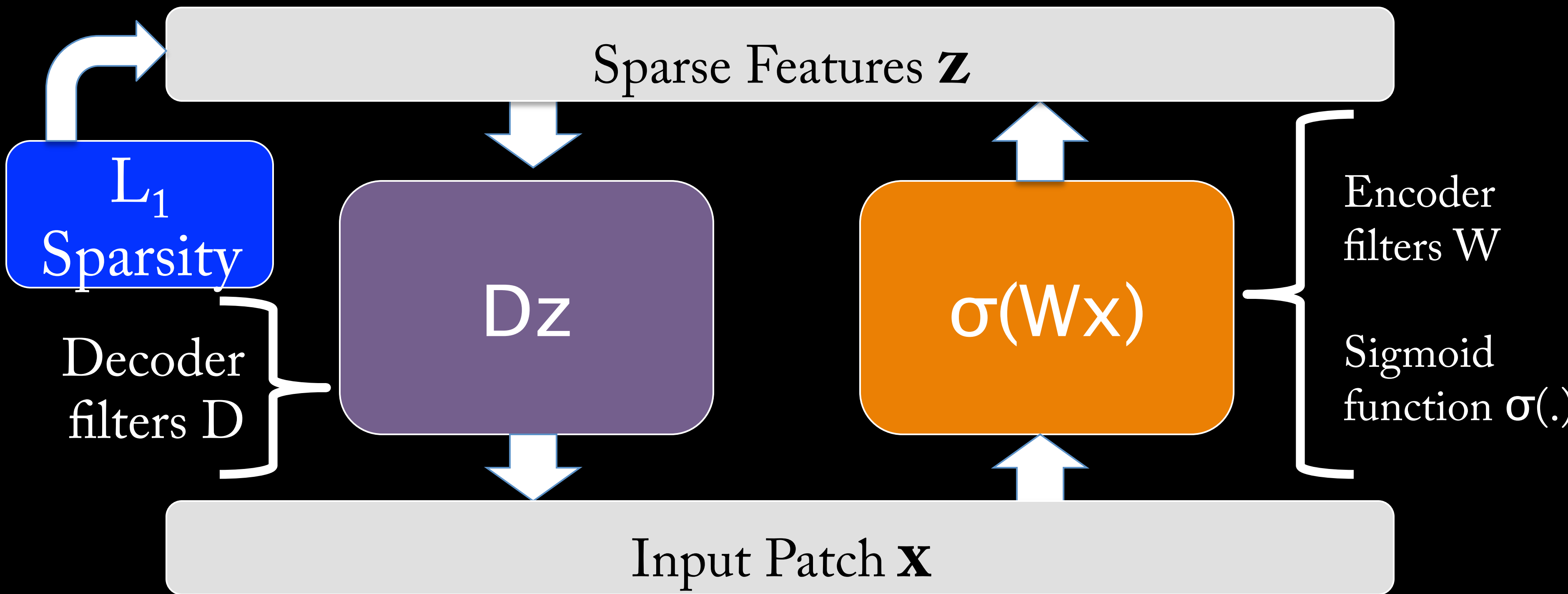
Auto-Encoder Example 2

- Predictive Sparse Decomposition [Ranzato et al., '07]



Auto-Encoder Example 2

- Predictive Sparse Decomposition [Kavukcuoglu et al., '09]



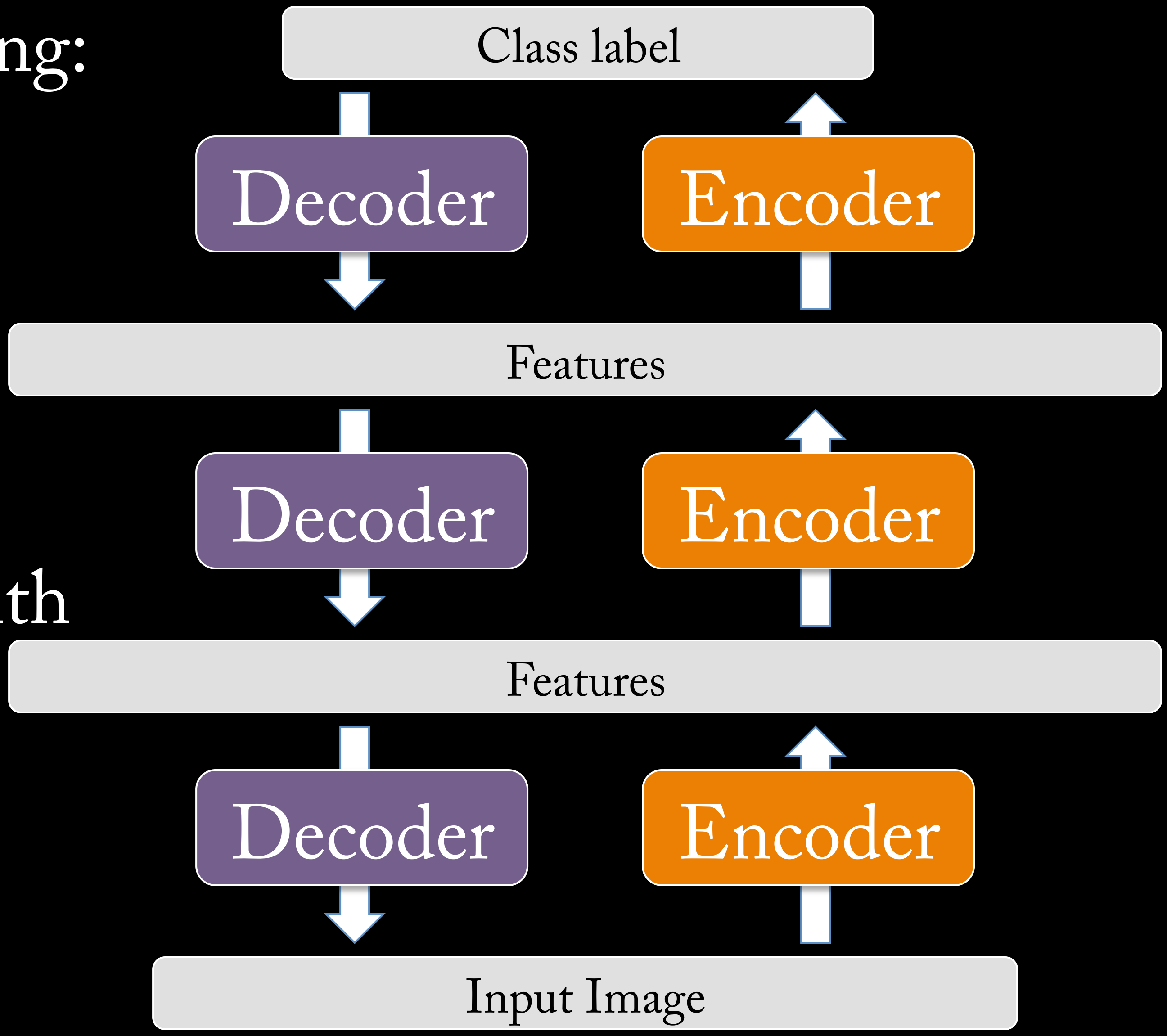
$$\min_{D,W,z} \underbrace{\|Dz - x\|_2^2 + \lambda|z|_1}_{Decoder} + \underbrace{\|\sigma(Wx) - z\|_2^2}_{Encoder}$$

Training

Stacked Auto-Encoders

Two phase training:

1. Unsupervised layer-wise pre-training
2. Fine-tuning with labeled data



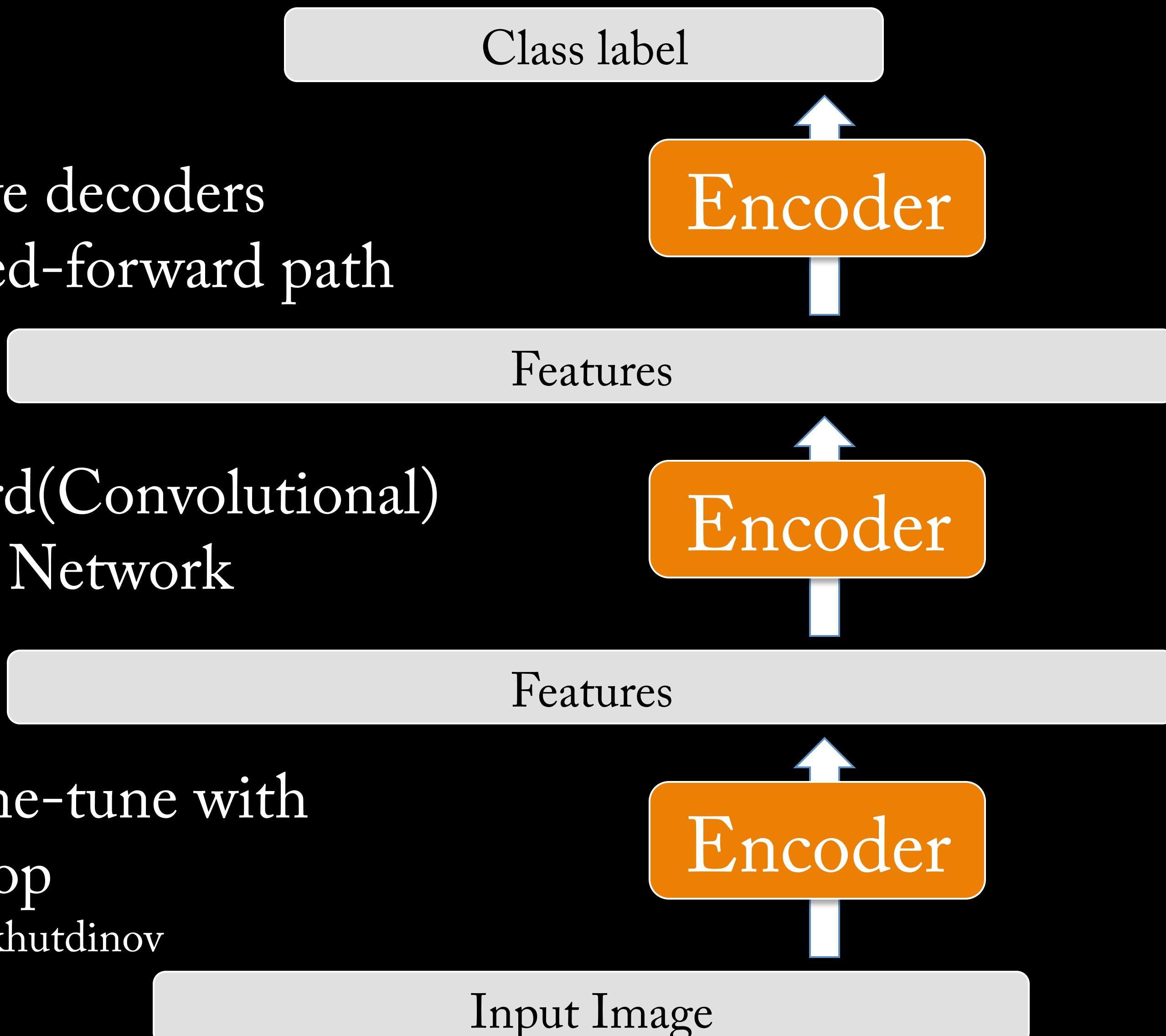
[Hinton & Salakhutdinov Science '06]

Training phase 2: Supervised Fine-Tuning

- Remove decoders
- Use feed-forward path

- Gives standard(Convolutional) Neural Network

- Can fine-tune with backprop

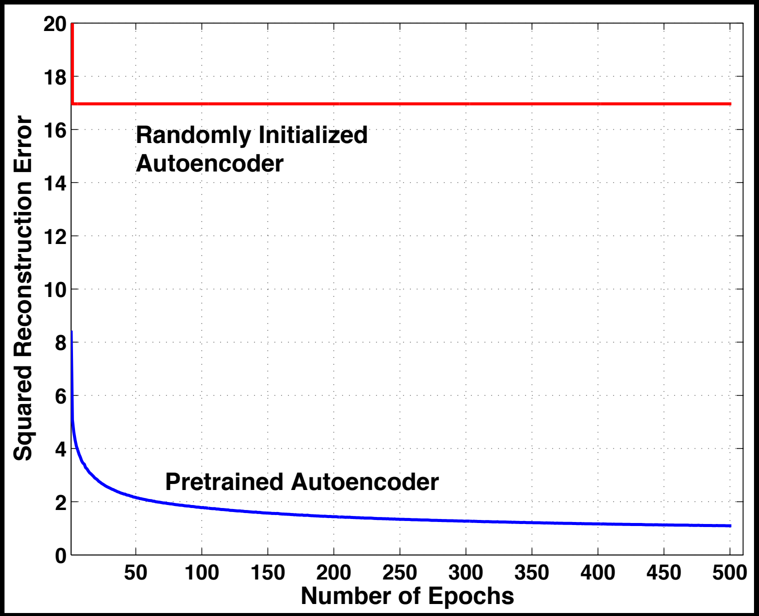


[Hinton & Salakhutdinov Science '06]

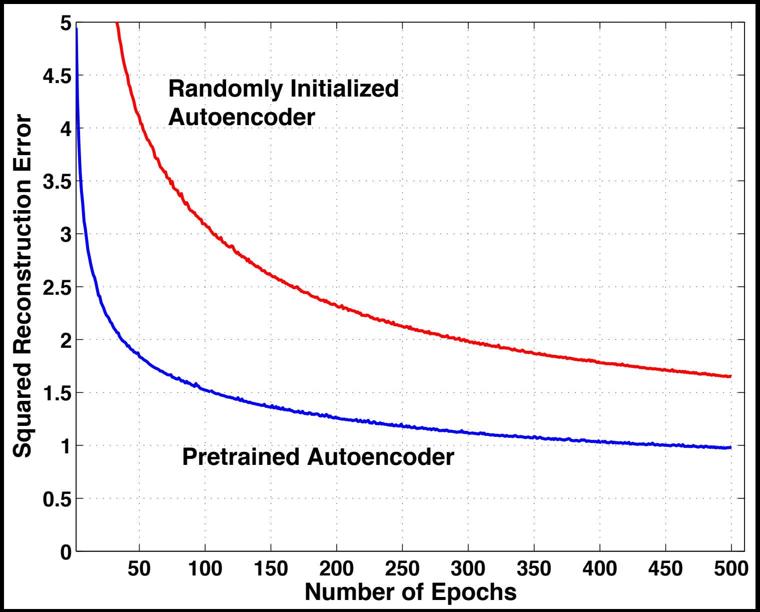
Effects of Pre-Training

- From [Hinton & Salakhudinov, Science 2006]

Big network



Small network

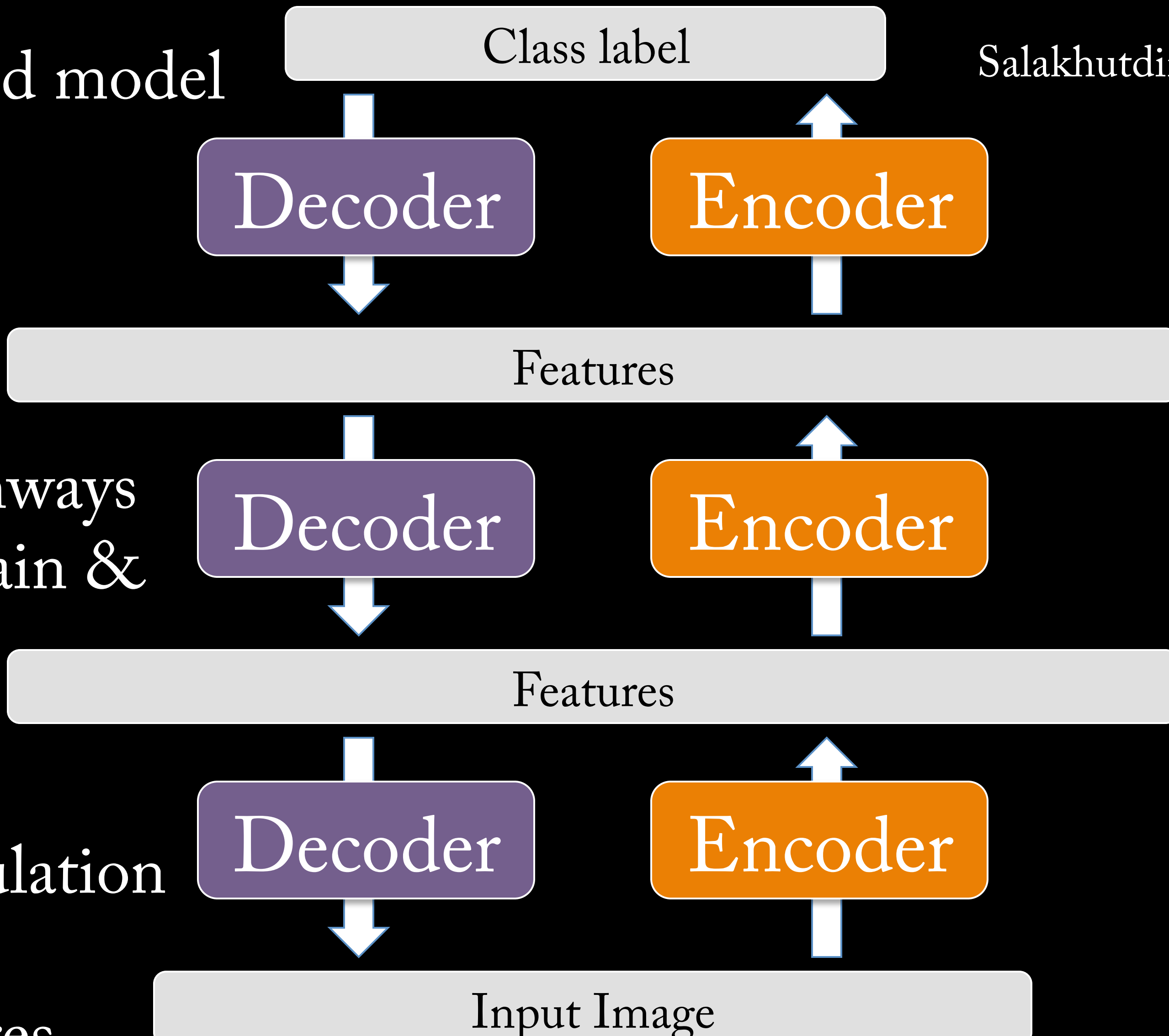


See also: Why Does Unsupervised Pre-training Help Deep Learning?
Dumitru Erhan, Yoshua Bengio ,Aaron Courville, Pierre-Antoine Manzagol
PIERRE-Pascal Vincent, Sammy Bengio, JMLR 2010

Deep Boltzmann Machines

Undirected model

Salakhutdinov & Hinton
AISTATS'09



Both pathways used at train & test time

TD modulation of
BU features

Shape Boltzmann Machine

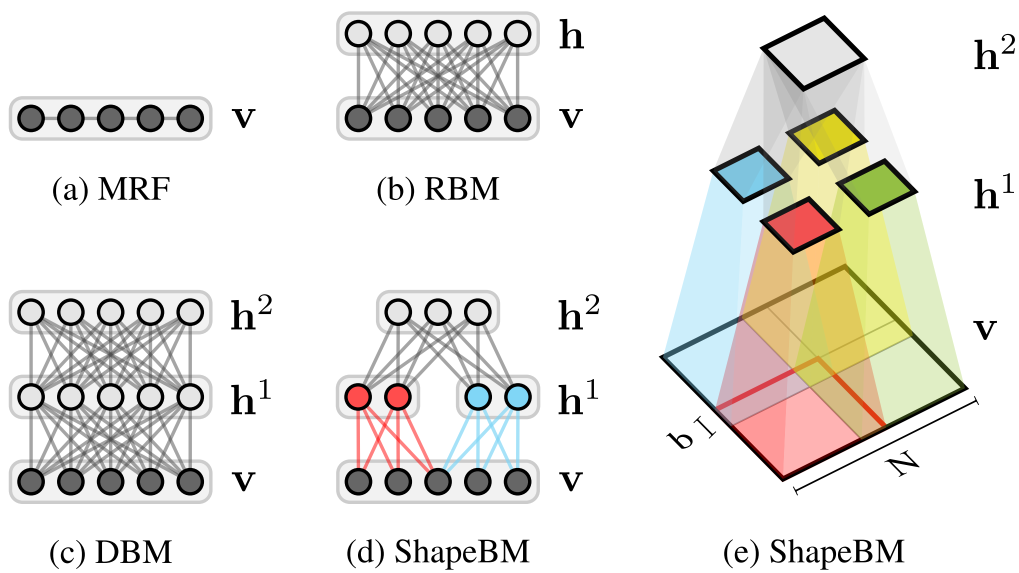
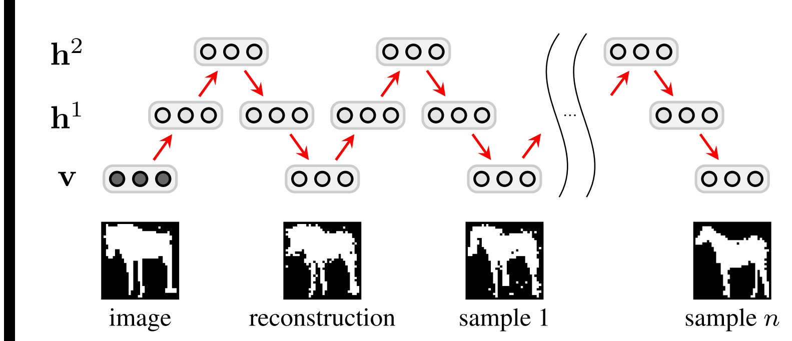
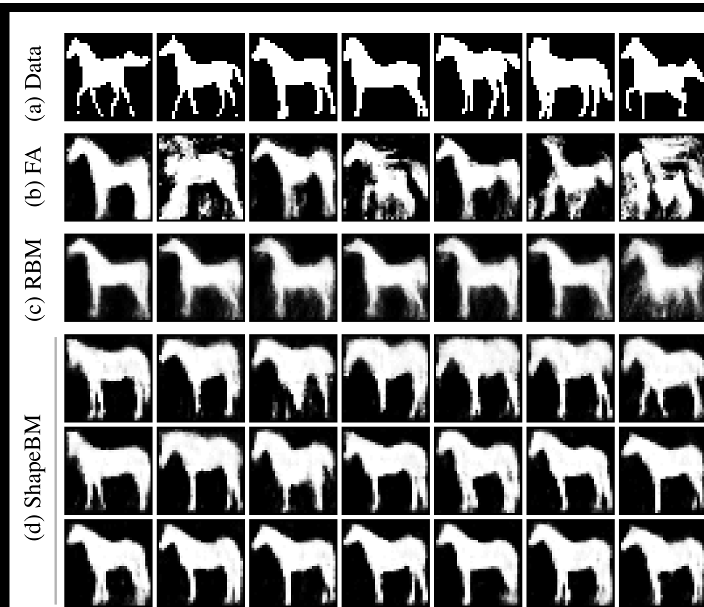


Figure 2. **Undirected models of shape:** (a) 1D slice of a Markov Random Field. (b) Restricted Boltzmann Machine in 1D. (c) Deep Boltzmann Machine in 1D. (d) 1D slice of a Shape Boltzmann Machine. (e) Shape Boltzmann Machine in 2D.

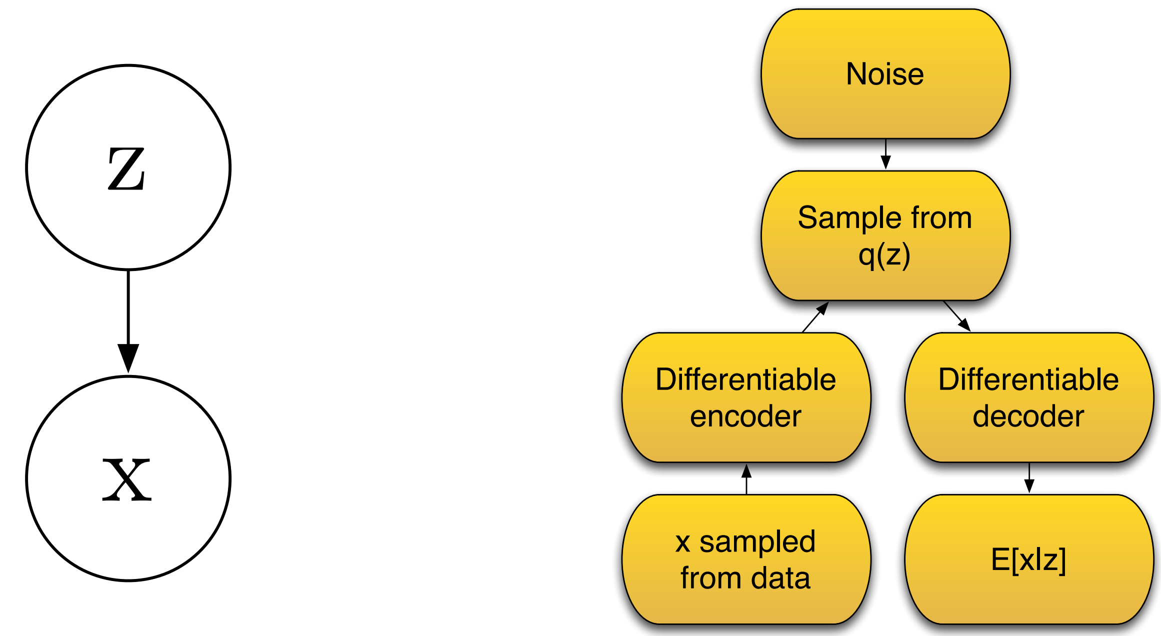




"The Shape Boltzmann Machine: a Strong Model of Object Shape", Ali Eslami, Nicolas Heess and John Winn, CVPR 2012

Variational Auto-Encoder

- [Kingma & Welling, ICLR 2014]



$$\text{Maximize} \log p(x) - D_{KL}\left(q(z \,\|p(z \mid x)\right)$$

(Kingma and Welling, 2014, Rezende et al 2014)

[Slide: Ian Goodfellow, Deep Learning workshop, ICML 2015]

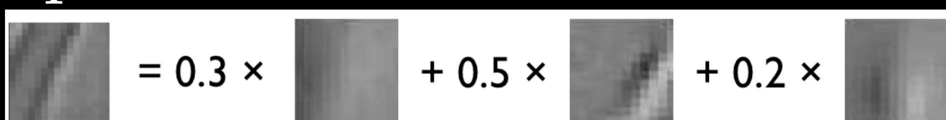
Decoder-Only Models

- Examples:
  - Sparse coding
  - Deconvolutional Networks [Zeiler & Fergus, '10]
- No encoder to compute features
- So need to perform optimization
  - Can be relatively fast

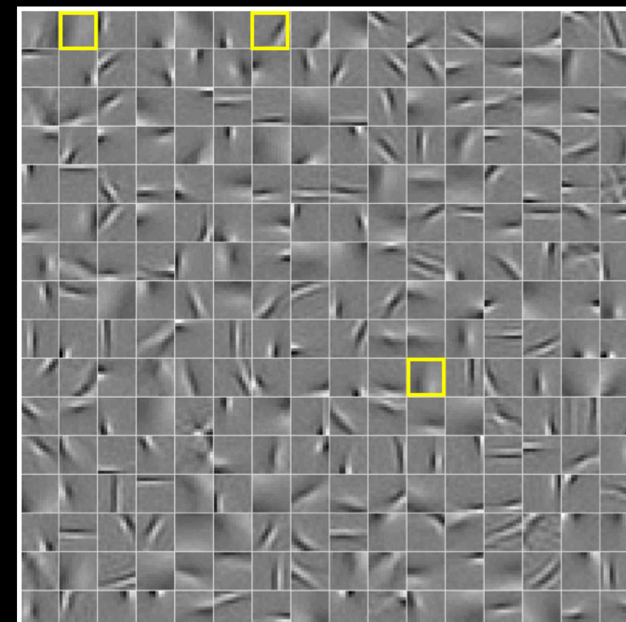
Sparse Coding (Patch-based)

- Over-complete linear decomposition of input y using dictionary D

Input



y



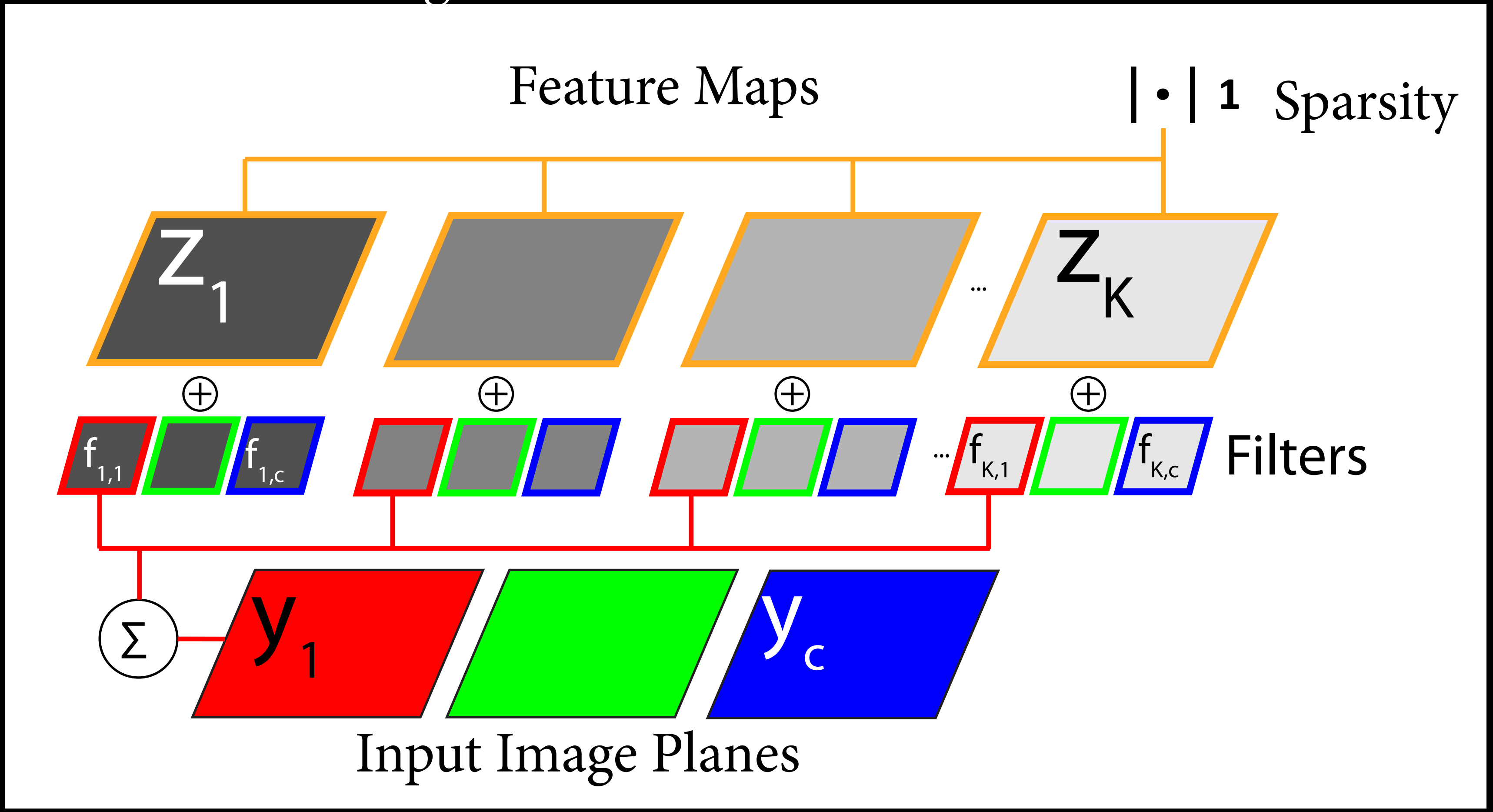
Dictionary D

$$C(y, D) = \underset{z}{\operatorname{argmin}} \ \frac{\lambda}{2} \|Dz - y\|_2^2 + |z|_1$$

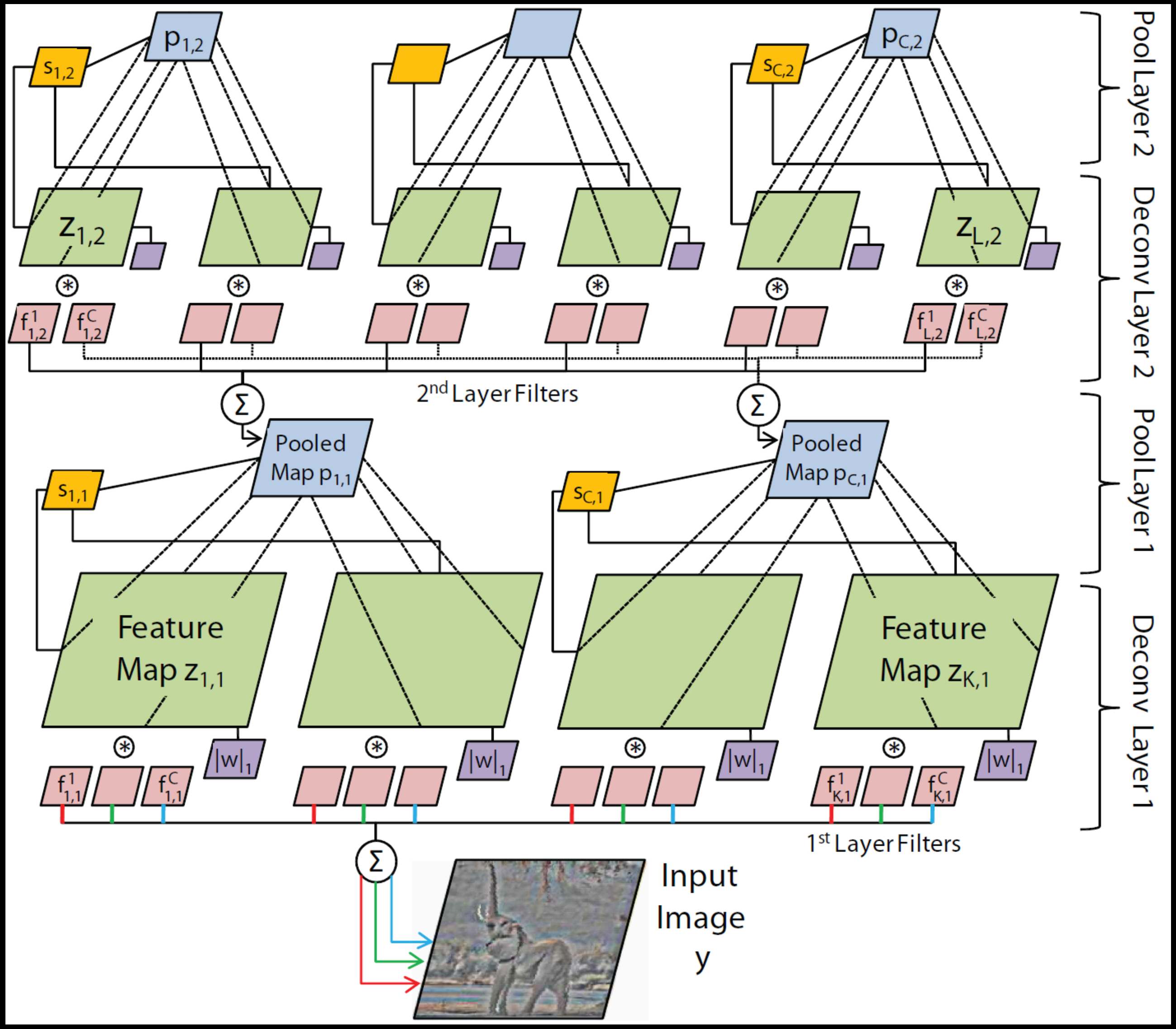
- ℓ_1 regularization yields solutions with few non-zero elements
- Output is sparse vector: $z = [0, 0.3, 0, \ldots, 0.5, \ldots, 0.2, \ldots, 0]$

Deconvolutional Network Layer

- Convolutional form of sparse coding
 [Zeiler & Fergus, CVPR 2010].
 Also Kavukcuoglu et al. NIPS 2010



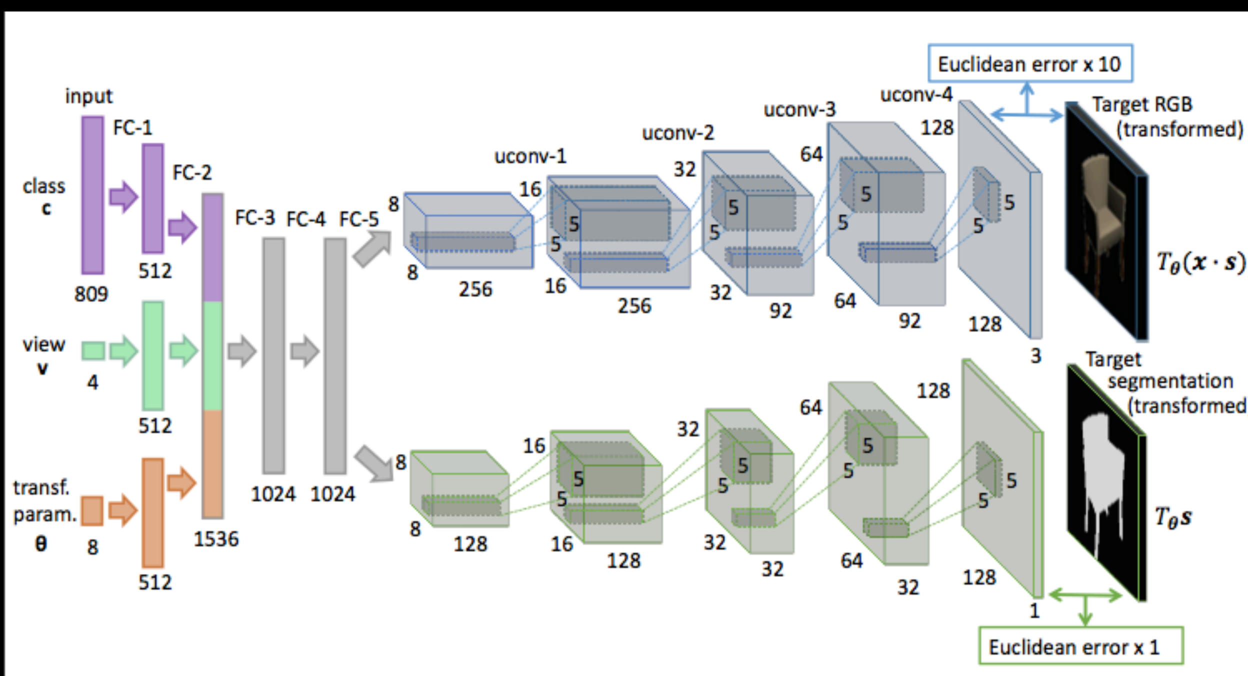
Overall Architecture (2 layers)

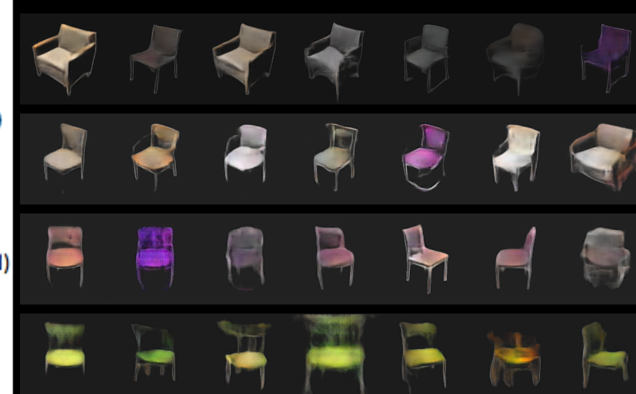


Generative Models using Convnets

- Learning to Generate Chairs with Convolutional Neural Networks, Alexey Dosovitskiy, Jost Tobias Springenberg and Thomas Brox, 1411.5928, 2014

- Supervised training of convnet to draw chairs

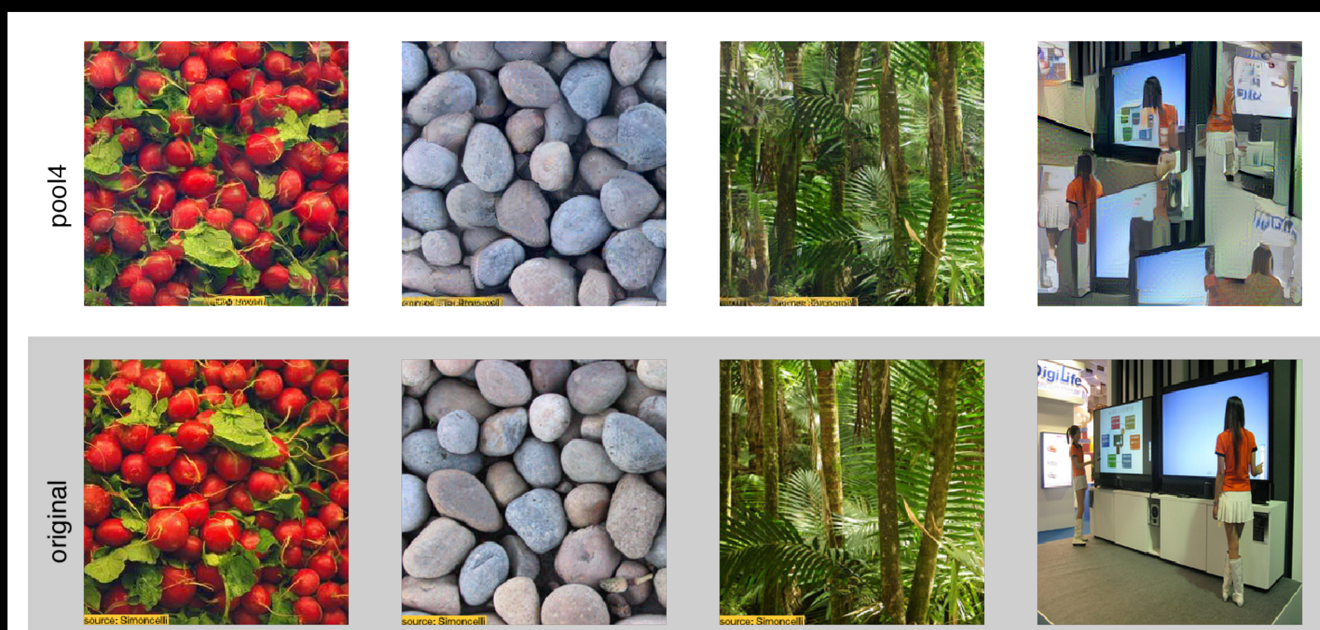




Some other interesting generative models

- "Generative Image Modeling Using Spatial LSTMs", L. Theis and M. Bethge, arXiv 1506.03478, 2015

- "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks", Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, . arXiv:1505.07376, 2015



Encoder-Only Models

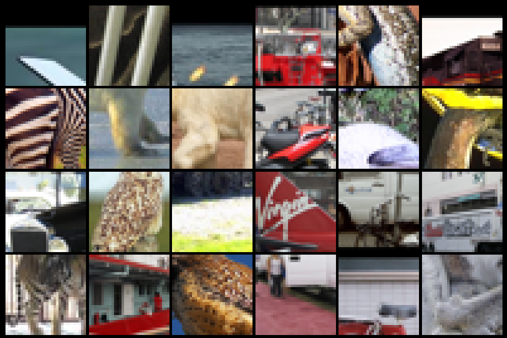
- In vision setting, essentially a convnet trained without explicit class labels

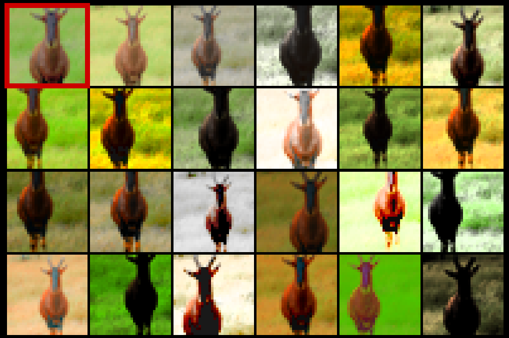
- Learn invariances
  - Unsupervised feature learning by augmenting single images, Alexey Dosovitskiy, Jost Tobias Springenberg and Thomas Brox, NIPS 2014

- Learn from video
  - Unsupervised Learning of Visual Representations using Videos Xiaolong Wang, Abhinav Gupta, arXiv 1505.00687, 2015

Unsupervised Learning of Transformations

[Unsupervised feature learning by augmenting single images, Alexey Dosovitskiy, Jost Tobias Springenberg and Thomas Brox, NIPS 2014]



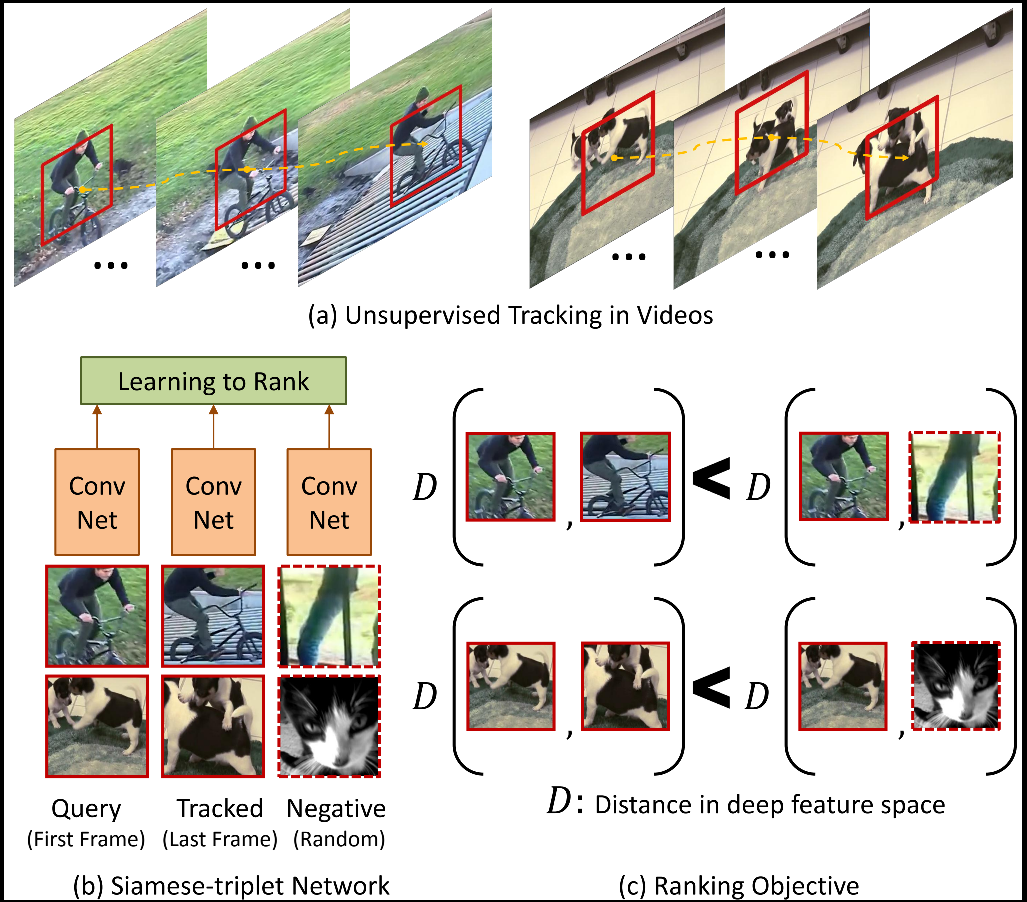


- Take patches from images
- For each patch, make lots of peturbed versions
- Treat each patch + peturbed copies as a separate classs
- Train supervised convnet

	STL-10	CIFAR-10-reduced	CIFAR-10	Caltech-101
K-means [6]	60.1 ± 1	70.7 ± 0.7	82.0	—
Multi-way local pooling [5]	—	—	—	77.3 ± 0.6
Slowness on videos [25]	61.0	—	—	74.6
Receptive field learning [16]	—	—	[83.11][1]	75.3 ± 0.7
Hierarchical Matching Pursuit (HMP) [3]	64.5 ± 1	—	—	—
Multipath HMP [4]	—	—	—	$\mathbf{82.5 \pm 0.5}$
Sum-Product Networks [8]	62.3 ± 1	—	[83.96][1]	—
View-Invariant K-means [15]	63.7	$\mathbf{72.6 \pm 0.7}$	**81.9**	—
This paper	$\mathbf{67.4 \pm 0.6}$	69.3 ± 0.4	77.5	76.6 ± 0.7 [2]

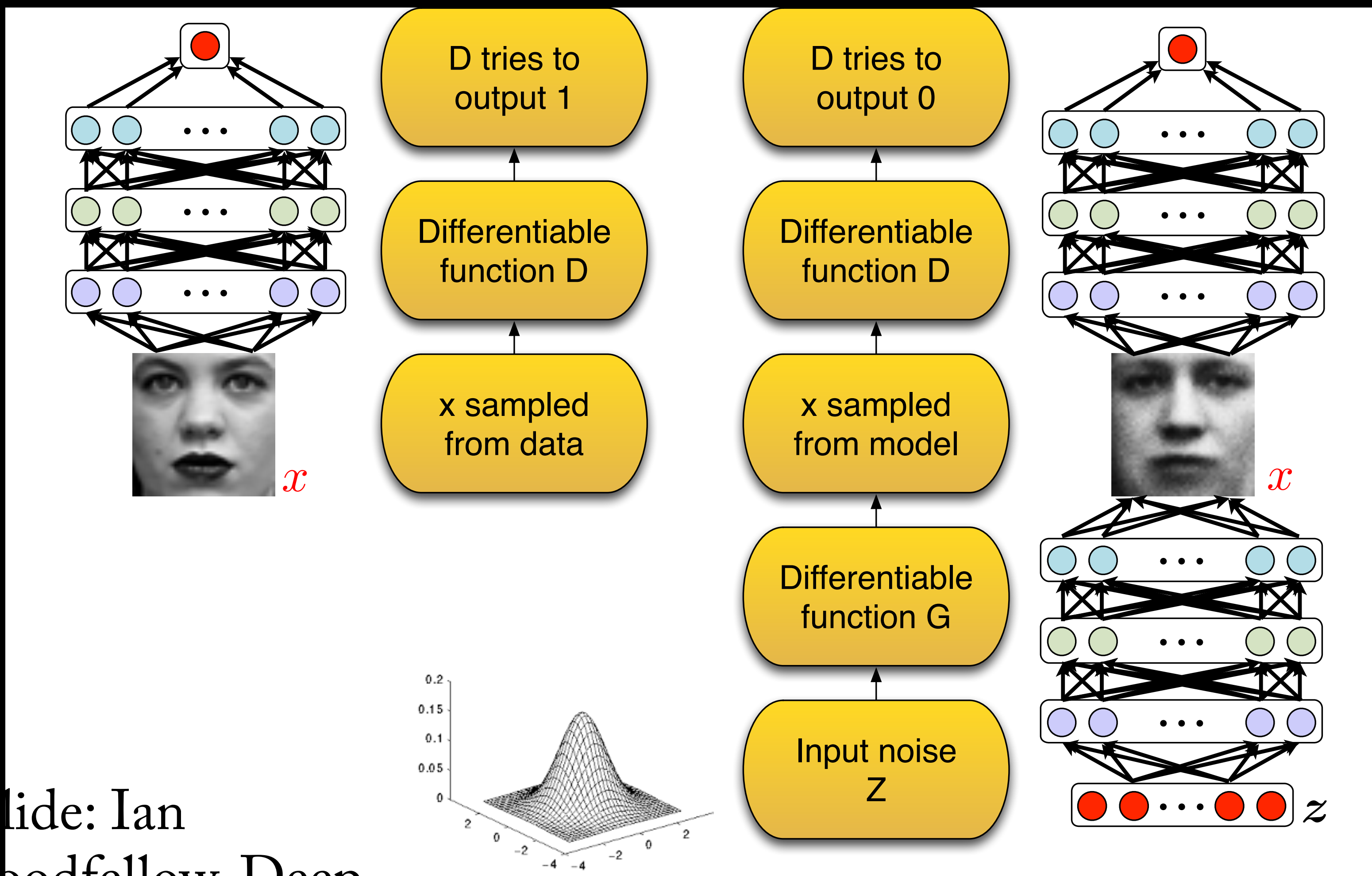
Unsupervised Learning from Video

- Unsupervised Learning of Visual Representations using Videos, Xiaolong Wang, Abhinav Gupta, arXiv 1505.00687, 2015



Generative Adversarial Networks

[Generative Adversarial Nets, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, NIPS 2014]



Generative Adversarial Networks

[Generative Adversarial Nets, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, NIPS 2014]

- Minimax value function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Generator pushes down

Discriminator pushes up

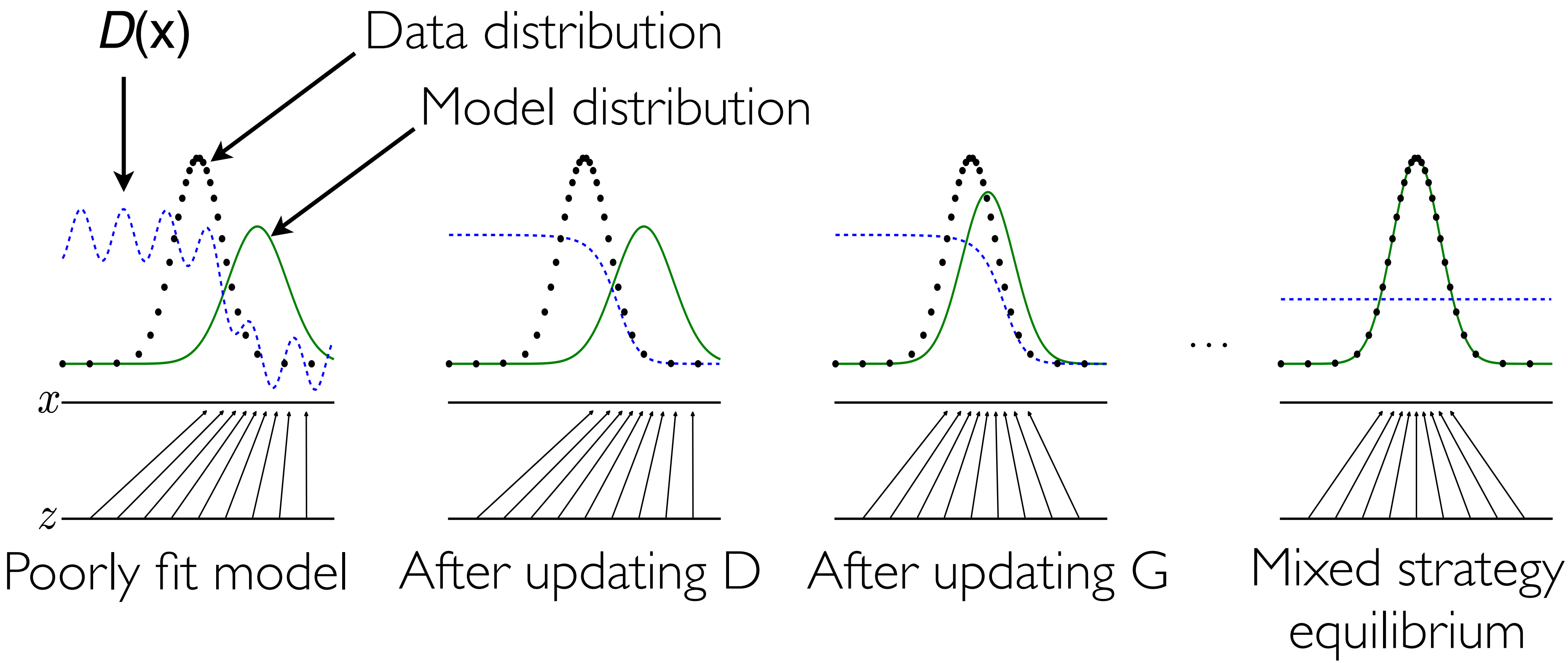
Discriminator's ability to recognize data as being real

Discriminator's ability to recognize generator samples as being fake

[Slide: Ian Goodfellow, Deep Learning workshop, ICML 2015]

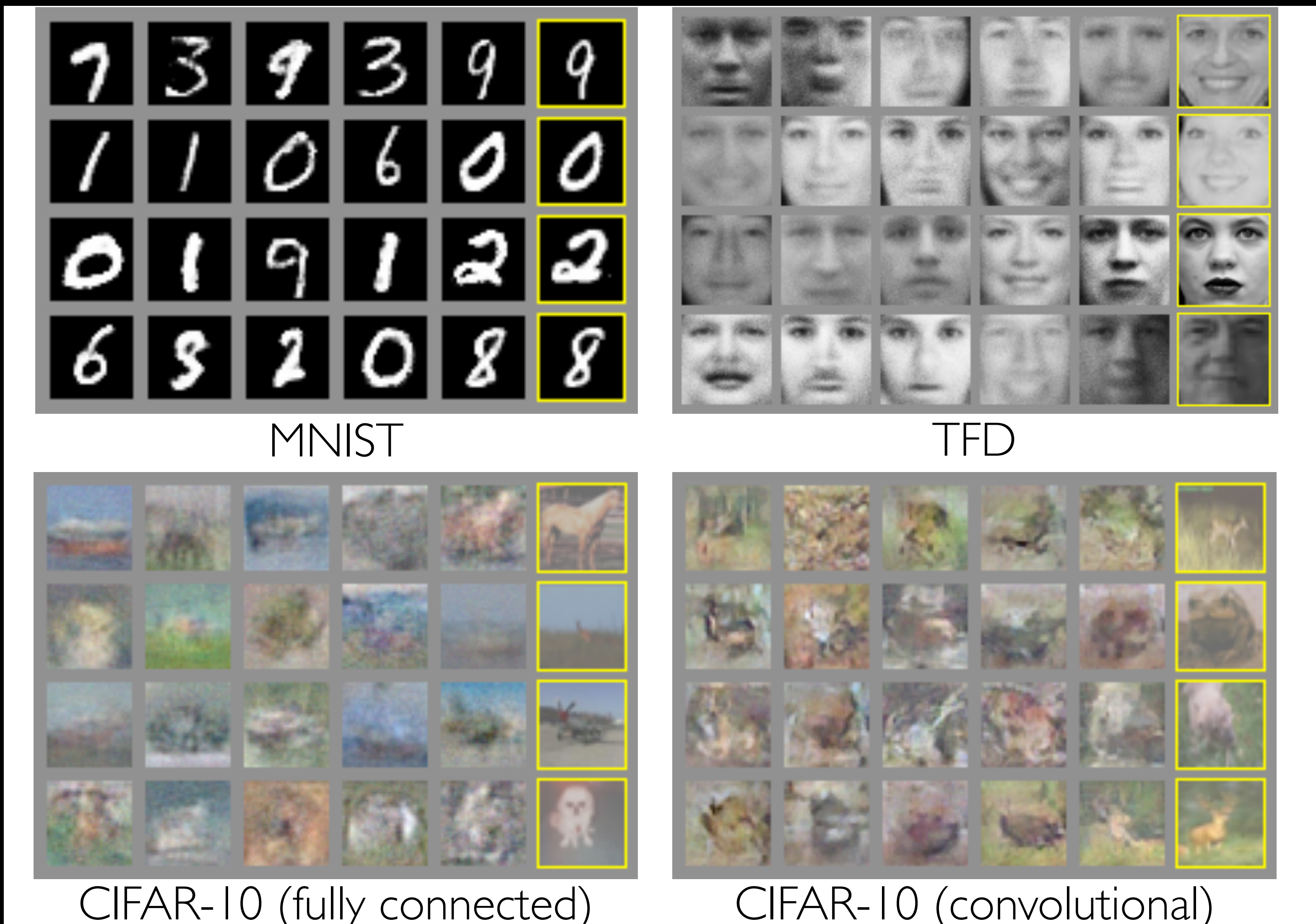
Generative Adversarial Networks

[Generative Adversarial Nets, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, NIPS 2014]



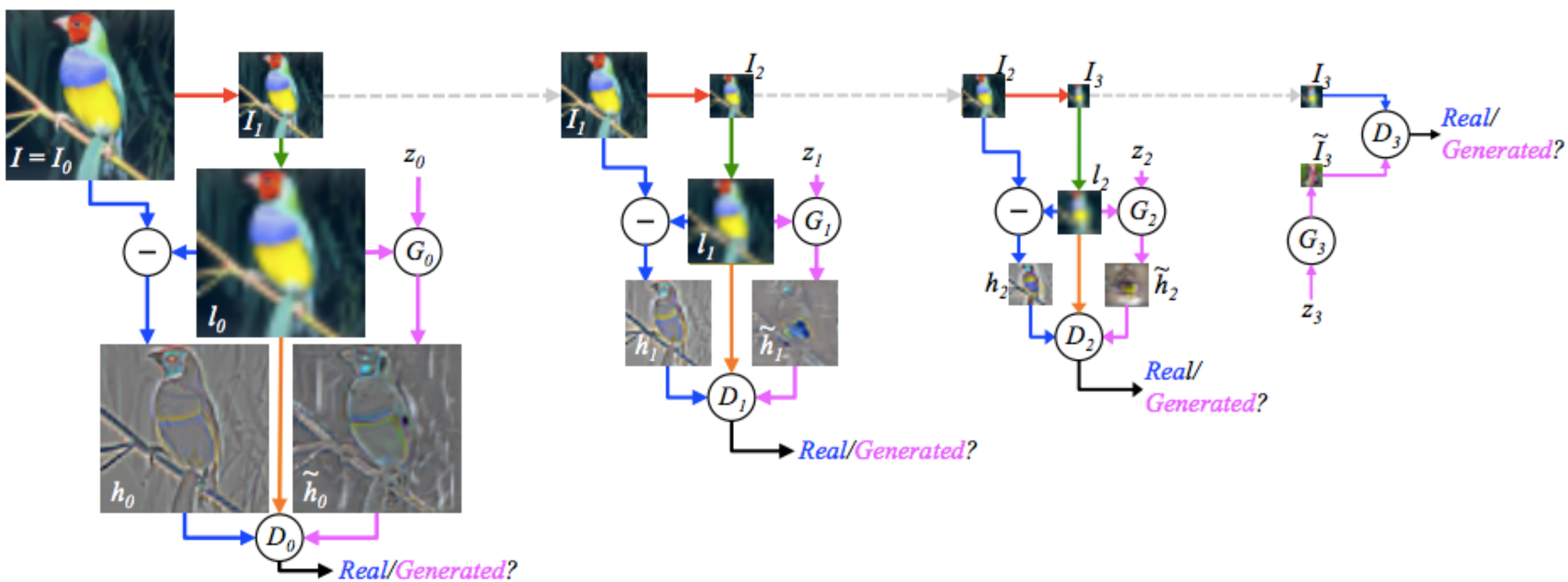
[Slide: Ian Goodfellow, Deep Learning workshop, ICML 2015]

Adversarial Network Samples



Adversarial Network using Laplacian Pyramid

- [Denton + Chintala, et al. arXiv 1506.05751, 2015]



Adversarial Network using Laplacian Pyramid

- [Denton + Chintala, et al. arXiv 1506.05751, 2015]

Memory in Neural Networks

Sainbayar Sukhbaatar

Introduction

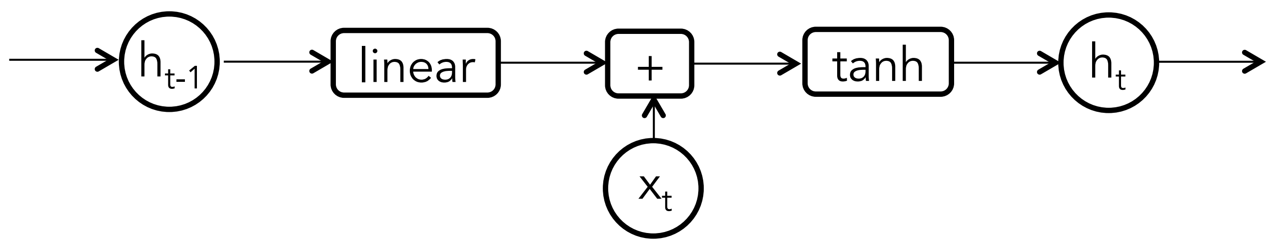
- Recently, there has been lot of interest in incorporating memory and attention to neural networks
  - Memory Networks, NTM, Learning to attend …
- Neural networks are not good at remembering things, especially when input is large but only part of it is relevant
- Adding external memory and learning to attend on important part is key

Outline

- Implicit Internal memory
  - RNN, LSTM
- Explicit External memory
  - MemNN, NTM
- Attention models
  - MT, Speech, Image, Pointer Network

Implicit Internal Memory

- Internal state of the model can be used for memory
  - Recurrent Neural Networks (RNNs)



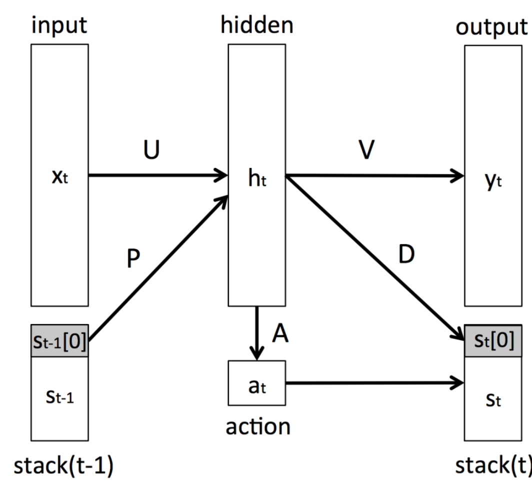
- Computation and memory is mixed
  - Complex computation requires many layers of non-linearity
  - But some information is lost with each non-linearity
  - Gradient vanishing, Catastrophic forgetting

Ways to Prevent Forgetting in RNNs

- Split state into fast and slow changing parts: structurally constrained recurrent nets (Mikolov et al., 2014)
  - Fast changing part is good for computation
  - Slow changing part is good for storing information
- Gated units for internal state
  - Control when to forget/write using gates
  - Long-short term memory (LSTM) (see Graves, 2013)
  - Simpler Gated Recurrent Unit (GRU) (Cho et al., 2014)
- Other problems
  - Memory capacity is fixed and limited by the dimension of state vector (computation is $O(N^2)$ where N is memory capacity)
  - Vulnerable to distractions in inputs
  - Restricted to sequential inputs

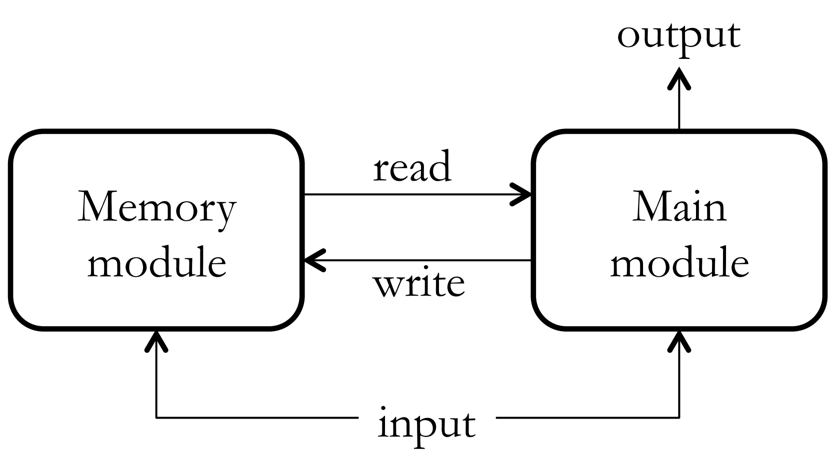
Stack memory for RNN (Joulin et al., 2014)

- Added a stack module to RNN, which can hold a list of vectors
- Action on stack: push, pop and no-op
- More powerful with multiple stacks
- Stack are updated in continuous manner → differentiable → trainable by backpropagation + search
- Applied to counting, memorization, binary addition



External Global Memory

- Separate memory from computation
  - Add separate memory module for storage
  - Memory contains list/set of items



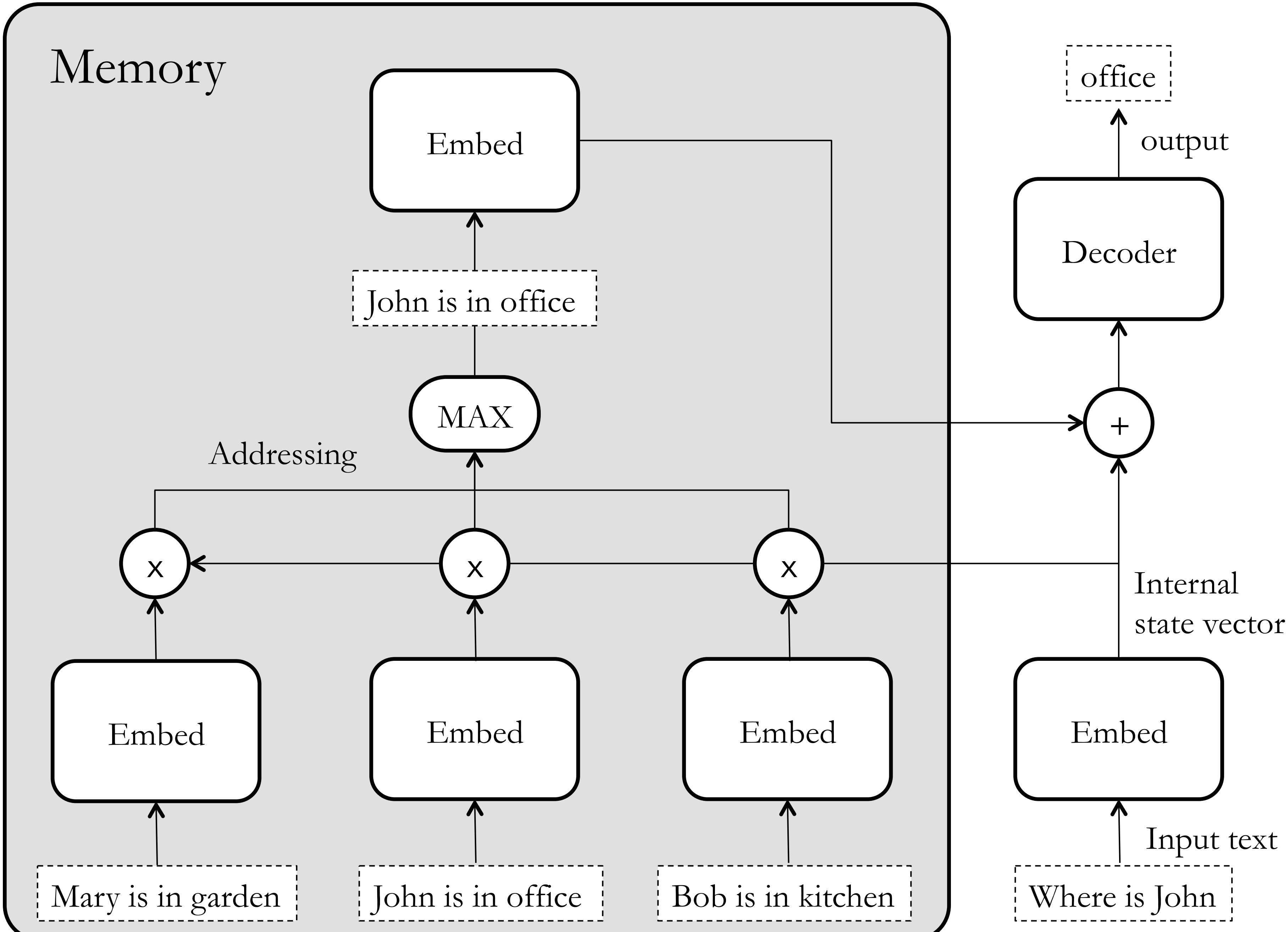
- Main module can read and write to the memory
- Advantage: long-term, scalable, flexible

Selective Addressing is Key for Memory

- Often, you only want to interact with few items in memory at once
  - Memory needs some addressing mechanism
- Memory addressing types
  - Soft or hard addressing
    - Soft addressing can be trained by backpropagation
    - Hard addressing is not differentiable (e.g. can be trained with reinforcement learning or additional training signal for where to attend)
  - Context and Location based addressing
    - When input is ordered in some way, location based addressing is useful
    - Location addressing is same as context if location is embedded in the context (e.g. MemN2N)

Memory Networks
(Weston et al., 2014)

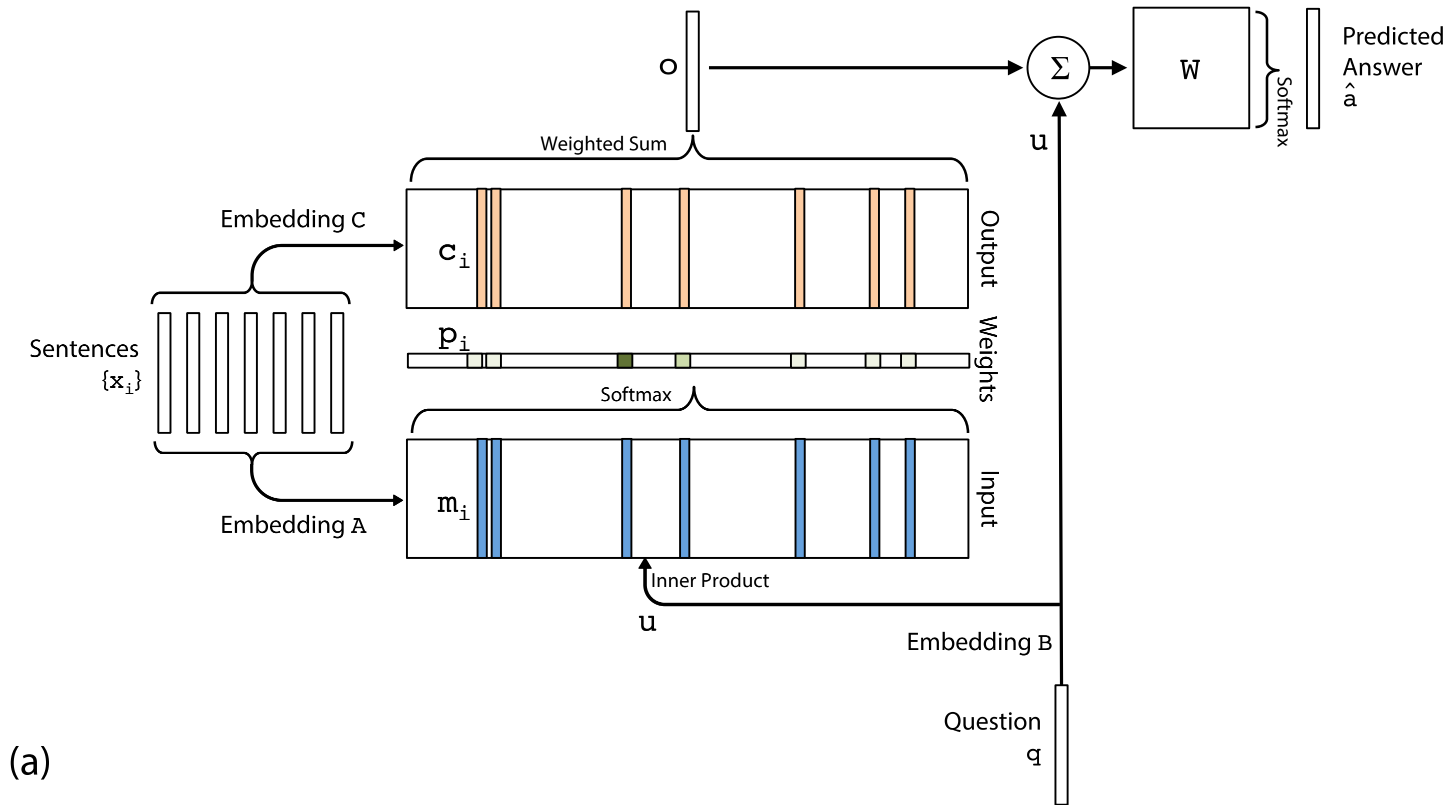
- Neural network with large external memory
- Writes everything to the memory, but reads only relative information
- Hard addressing: max of the inner product between then internal state and memory contents
- Location based addressing: can compare two memory items by their relative location
- Can perform multiple memory lookups (hops) before producing an output
- Requires additional training signals for training hard addressing
- Applied to toy and large-scale QA tasks


Memory
Embed
office
output
Decoder
John is in office
MAX
+
Addressing
x
x
x
Internal
state vector
Embed
Embed
Embed
Embed
Input text
Mary is in garden
John is in office
Bob is in kitchen
Where is John

End-to-end Memory Networks (Sukhbaatar et al., 2015)

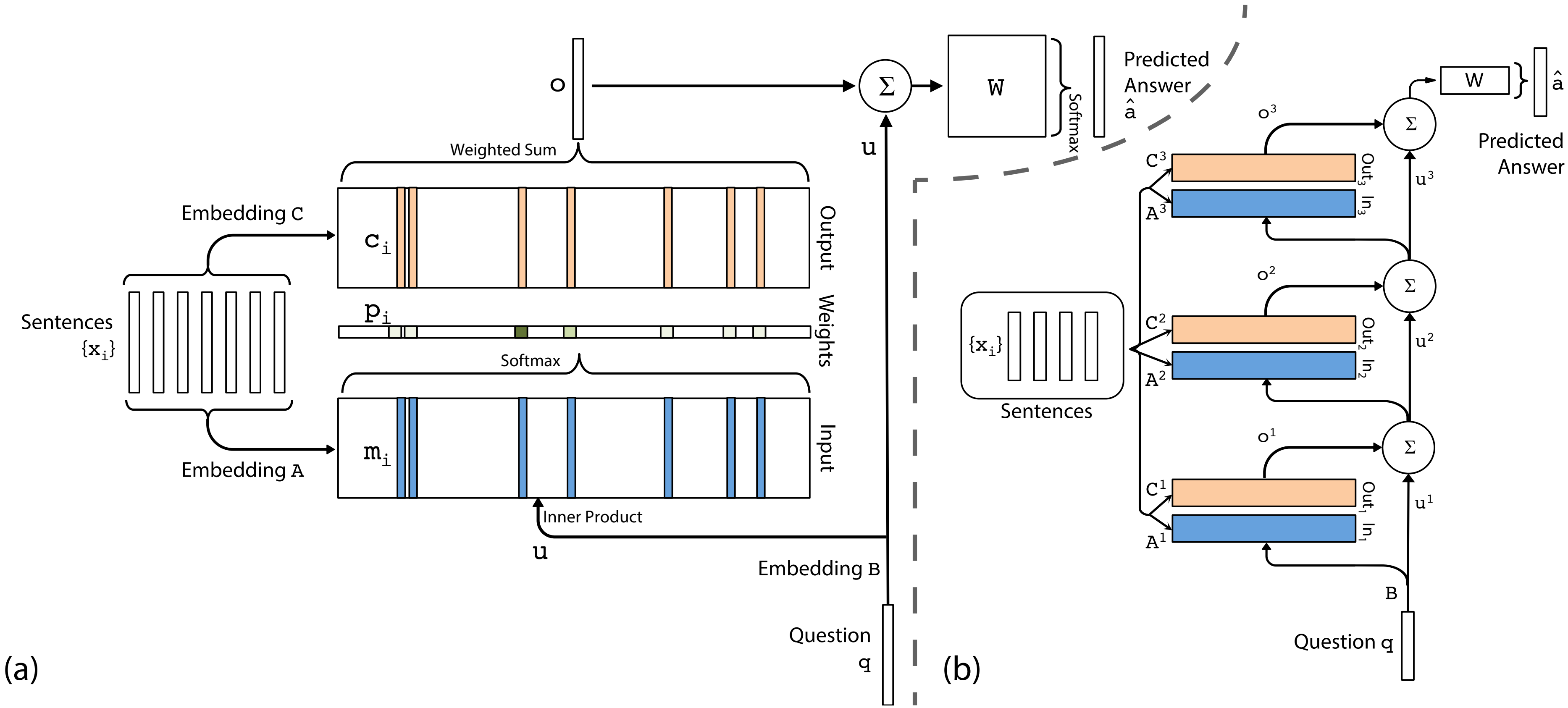
- Soft addressing: replaced hard max with **softmax**
- End-to-end training: softmax is differentiable → can train with backpropagation
- Location addressing: location/time is embedded into the context (special words for "Time=4")
- Applied to toy QA and language modeling

End-to-end Memory Networks (Sukhbaatar et al., 2015)



Single Memory Lookup

End-to-end Memory Networks (Sukhbaatar et al., 2015)

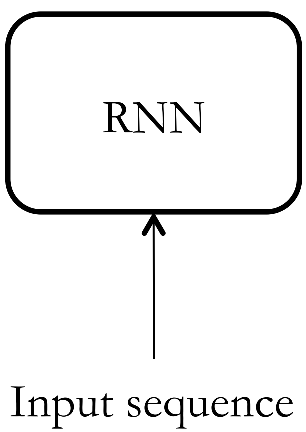


Single Memory Lookup

Multiple Memory Lookup

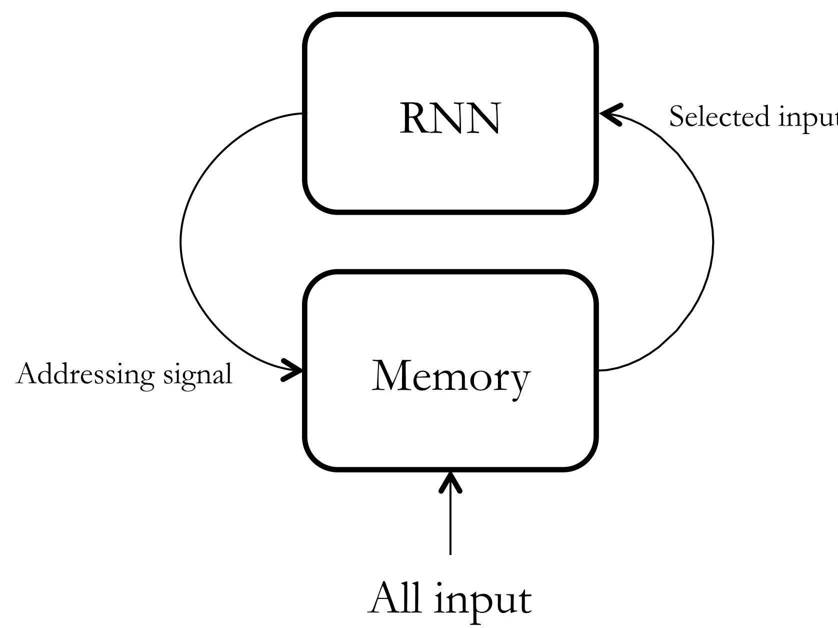
RNN viewpoint of End-to-End MemNN

Plain RNN



Inputs are fed to RNN one-by-one in order. RNN has only one chance to look at a certain input symbol.

Memory Network



Place all inputs in the memory. Let the model decide which part it reads next.

Attention during memory lookups

Samples from toy QA tasks (bAbI dataset)

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk? Answer: hallway Prediction: hallway				

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

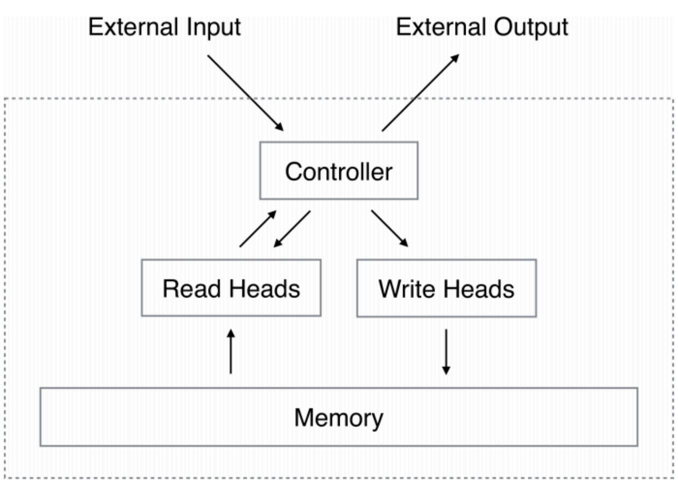
Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

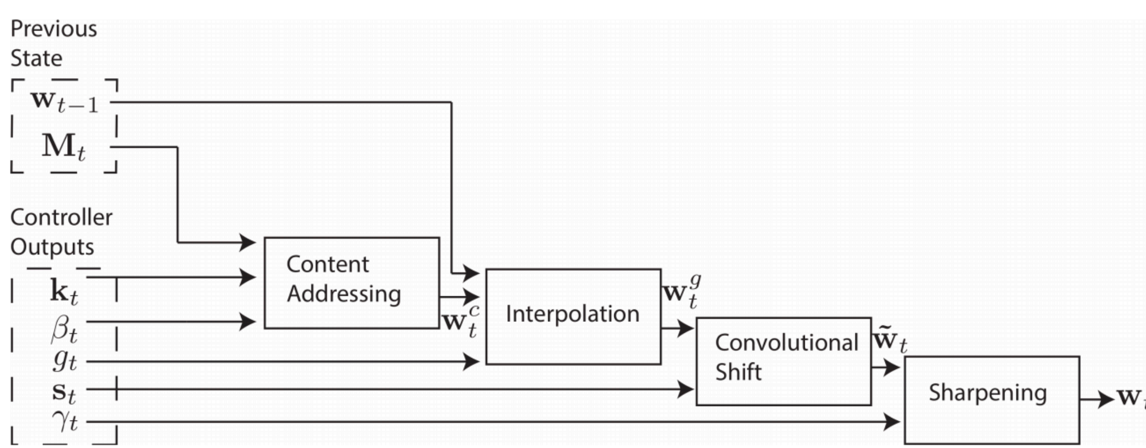
Result

	Test error	Failed tasks
MemNN	6.7%	4
LSTM	51%	20
MemN2N	12.4%	11

Neural Turing Machine (Graves et al., 2014)

- Learns how to write to the memory
- Soft addressing → backpropagation training
- Location addressing: small continuous shift of attention
- Complex addressing mechanism: need to sharpen after convolution
- Controller can be LSTM-RNN or feed-forward neural network
- Applied to learn algorithms such as sort, associative recall and copy.
- Hard addressing with reinforcement learning (Zaremba et al., 2015)

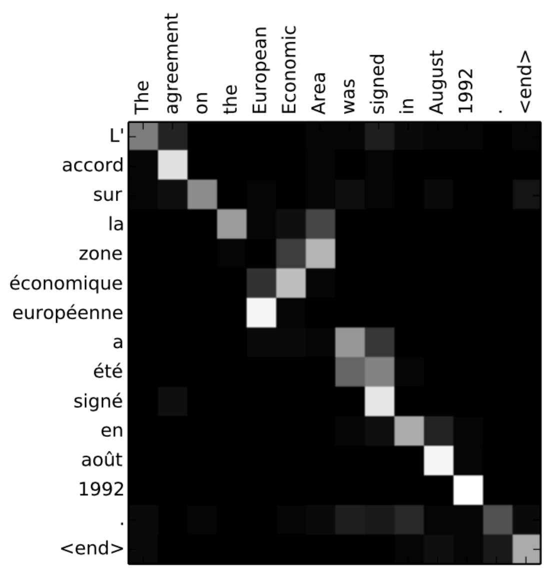




RNNsearch: Attention in Machine Translation (Bahdanau et al., 2015)

- RNN based encoder and decoder model
- Decoder can look at past encoder states using soft attention
- Attention mechanism is implement by a small neural network
  - It takes the current decoder state and a past encoder state and outputs a score. Then the all scores are fed to softmax to get attention weights
- Applied to machine translation. Significant improvement in translation of longer sentences

Attention weights during English to French machine translation



Significant improvement on long sentences

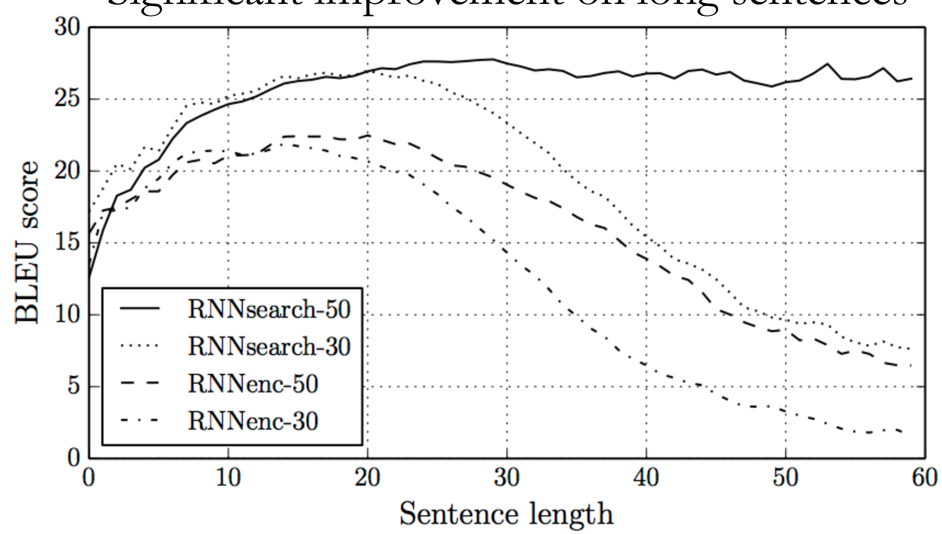
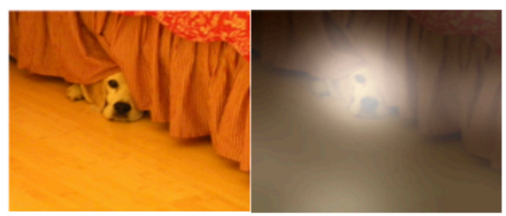


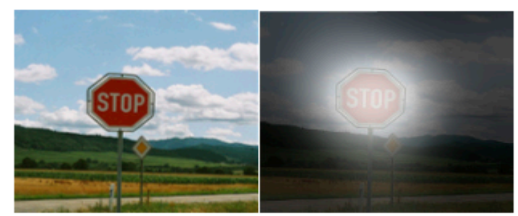
Image caption generation with attention (Xu et al., 2015)

- Encoder: lower convolutional layer of a deep ConvNet (because need spatial information)
- Decoder: LSTM RNN with soft spatial attention
  - Decoder state and encoder state at single location are fed to small NN to get score at that location
- Network attends to the object when it is generating a word for it
- Also hard attention is tried with reinforcement learning

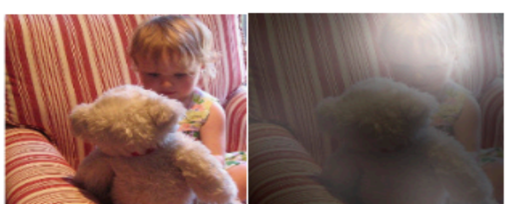
A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



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

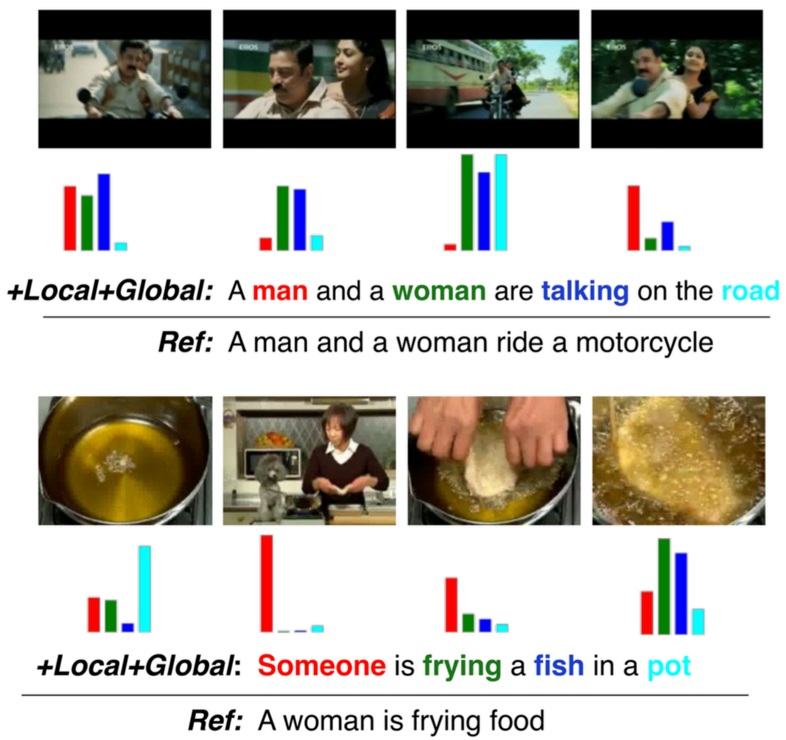


A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Video description generation (Yao et al., 2015)

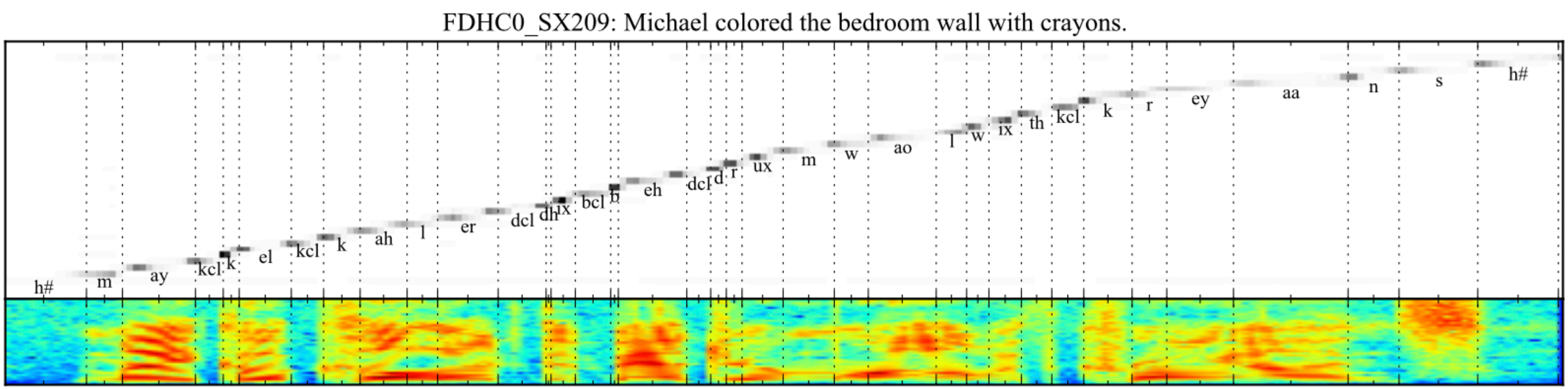


(bottom: ground truth)

L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” *arXiv: 1502.08029*, 2015.

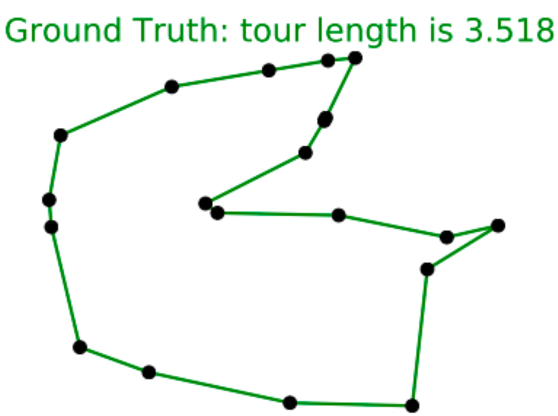
Location-aware attention for speech (Chorowski et al., 2015)

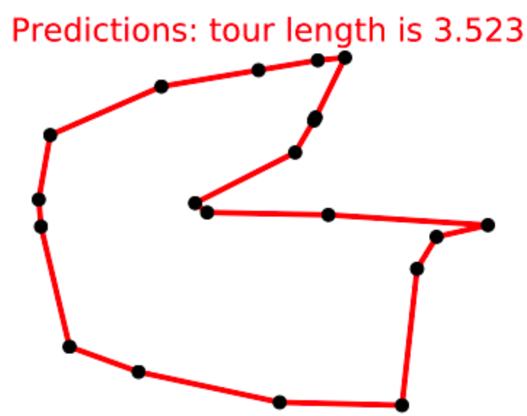
- RNN based encoder-decoder model with attention (similar to RNNsearch)
- Location based addressing: previous attention weights are used as feature for the current attention (good when subsequent attention locations are highly correlated)
- Improvement with sharpening and smoothing of memory addressing



Pointer Network: attention as an output (Vinyals et al., 2015)

- RNN based encoder-decoder model for discrete optimization problems
- Decoder can attend to previous encoder states (similar to RNNsearch, content based soft attention by a small NN)
- Rather than fixed output classes, attention weights determine output
- Input to the most attended encoder state becomes an output → can output any sequence of inputs





Resources

- EMNLP 2014 tutorial
  - http://emnlp2014.org/tutorials.html#embedding
- CVPR2014 deep learning tutorial
  - https://sites.google.com/site/deeplearningcvpr2014/
- ICML 2013 deep learning tutorial
  - http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf

Software

- Caffe (http://caffe.berkeleyvision.org/)
  - Vision-centric
- Torch (http://torch.ch/)
  - Lua-based library for Deep Learning
  - Currently used by FAIR and Google Deep Mind
- Theano (http://deeplearning.net/software/theano/)
  - Automatic differentiation
  - Python-based

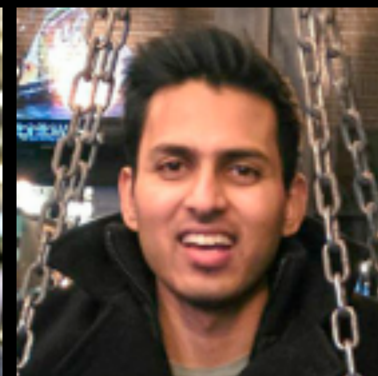
Thanks!

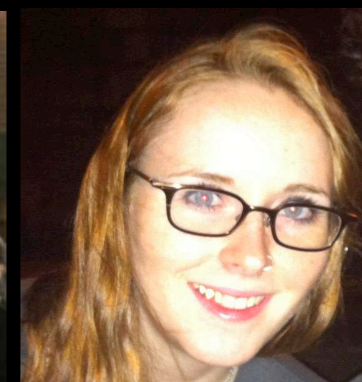
Facebook AI Research colleagues & NYU PhD students:

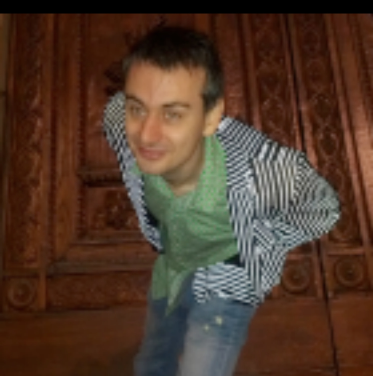

Manohar Paluri


Antoine Bordes


Yaniv Taigman


Soumith Chintala


Emily Denton


Jason Weston


Tomas Mikolov


Ronan Collobert


Marc'Aurelio Ranzato


Sainbayar Sukhbaatar

FAIR Overview

Facebook AI Research

- Toward Artificial Intelligence (AI), with **Machine Learning**.
- Established Dec 2013 (1.5 year old)
  - initiative of CEO and CTO
  - lead by **Yann Lecun**

FAIR Overview

Facebook AI Research

- ~35 researcher scientists
  - Machine Learning, Computer Vision and Natural Language Processing
- ~15 research engineers
  - Software support, prototyping, interaction with product teams…
- Locations:
  - New York City
  - Menlo Park (HQ)
  - Paris

FAIR Mission

Facebook AI Research

- Advance the state of the art of AI
  - Publish research in best conferences and journals
  - Open-source code release
- Produce software tools for AI research and applications
- Help FB products to leverage advances in AI
  - Software prototyping, architecting, interaction with product teams…

FAIR Impact

Machine Learning @ FB

- Computer Vision
  - Face detection and identification
  - Object detection, scene classification
  - Video classification
- Natural Language
  - Tag prediction for search, feed ranking, ad targeting
- Computational Advertising
  - Ads targeting
  - User interest modeling

Huge Scale Deployment of Machine Learning

- 1.4 billion monthly active users
- 850 million daily active users (1 in 7 people on Earth)
- More images uploaded than any other website
  - 400M+ new Facebook photos/day (no labels)
  - 60M+ Instagram images/day (most with hashtags)
  - ~ 500 Billion photos total
- Face and Object recognition models applied to <u>every</u> image
- 5M video uploads/day & growing rapidly
  - More video playback than YouTube

We are hiring!

- Internships
  - https://www.facebook.com/careers/department?dept=grad&req=a0IA000000CzCGuMAN
- Postdoc positions
  - Ex-postdocs now faculty at Berkeley, Harvard
- Full-time positions

- https://research.facebook.com/ai

Memory References

- A. Graves. Generating sequences with recurrent neural networks. arXiv preprint: 1308.0850, 2013
- T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. A. Ranzato. Learning longer memory in recurrent neural networks. arXiv preprint:1412.7753, 2014
- A. Joulin, and T. Mikolov. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. arXiv preprint:1503.01007, 2015
- K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint:1409.1259, 2014
- J. Weston, S. Chopra, and A. Bordes. Memory networks. In International Conference on Learning Representations (ICLR), 2015
- S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-To-End Memory Networks. arXiv preprint:1503.08895, 2015

Memory References

- A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. arXiv preprint: 1410.5401, 2014
- W. Zaremba, and I. Sutskever. Reinforcement Learning Neural Turing Machines. arXiv preprint:1505.00521, 2015
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR), 2015
- K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. ICML, 2015
- L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. arXiv preprint: 1502.08029, 2015
- J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. arXiv preprint: 1506.07503, 2015
- O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. arXiv preprint:1506.03134, 2015

Neuroscience of memory

- hippocampus
  - Densely connected
  - Vital for new memory formation
  - From few days to few years
  - Place / grid cells
- Neo-cortex
  - Can keep memory much longer

Memory types

- Short-term memory (working memory)
  - Limited capacity
- Long term memory
  - Explicit / Declarative
    - Semantic memory
    - Episodic memory
  - Implicit
    - Procedural memory
    - Priming