Microsoft Research

Deep Learning

for Natural Language Processing and Related Applications

Xiaodong He, Jianfeng Gao, and Li Deng

Deep Learning Technology Center (DLTC), MSR, Redmond, WA, 98052

May, 2014

Microsoft Research

Tutorial Outline

- **Part I (by Li Deng): Background of deep learning, common and natural Language Processing (NLP) centric architectures**
    - Deep learning Background
        - Industry impact & Basic definitions
        - Achievements in speech, vision, and NLP
    - Common deep learning architectures and their **speech/vision** applications
        - Fully connected deep neural nets (DNN), DNN-HMM, CD-DNN-HMM, Tensor DNN
        - Deep convolutional neural nets (CNN)
        - Deep stacking networks (DSN), kernel DSN, tensor DSN, recurrent DSN
        - Recurrent neural nets (RNN), bi-directional RNN, deep RNN, LSTM-RNN
    - Deep learning architectures for modeling **NL** structure
        - Neural network & RNN for language modeling
        - Models for word embeddings
        - Recursive neural networks with local and global contexts
        - DSSM (Deep Structured Semantic Model; Deep Semantic Similarity Model) and its variants

- **Part II (by Xiaodong He): Deep learning in spoken language understanding (SLU)**
  - Overview of SLU
  - Domain & intent detection using DNN
  - Slot filling/sequential tagging using RNN
- **Part III (by Xiaodong He): Learning semantic embedding**
  - Word embedding and sub-word embedding
  - Semantic embedding: from word to phrase & document
  - Learning semantic embedding using DSSM
- **Part IV (by Jianfeng Gao): Deep learning in machine translation**
  - Overview of statistical machine translation
  - DNN-based semantic translation models
- **Part V (by Jianfeng Gao): Deep semantic similarity models**
  - Overview of semantic similarity models
  - Deep structured semantic models (DSSM) for Web Search

Microsoft Research

Part I

Background/Impact of Deep Learning

Common and NLP-centric architectures

MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2013

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

Temporary Social Media

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

Prenatal DNA Sequencing

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

Additive Manufacturing

Skeptical about 3-D printing? GE, the world's largest manufacturer, is on the verge of using the technology to make jet parts.

Baxter: The Blue-Collar Robot

Rodney Brooks's newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.

Memory Implants

A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.

Smart Watches

The designers of the Pebble watch realized that a mobile phone is more useful if you don't have to take it out of your pocket.

Ultra-Efficient Solar Power

Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.

Big Data from Cheap Phones

Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.

Supergrids

A new high-power circuit breaker could finally make highly efficient DC power grids practical.

The New York Times

Scientists See Promise in Deep-Learning Programs

John Markoff

November 23, 2012

Geoff Hinton

Rich Rashid in **Tianjin**, October, 25, 2012

Impact of deep learning in speech technology

MIT Technology Review

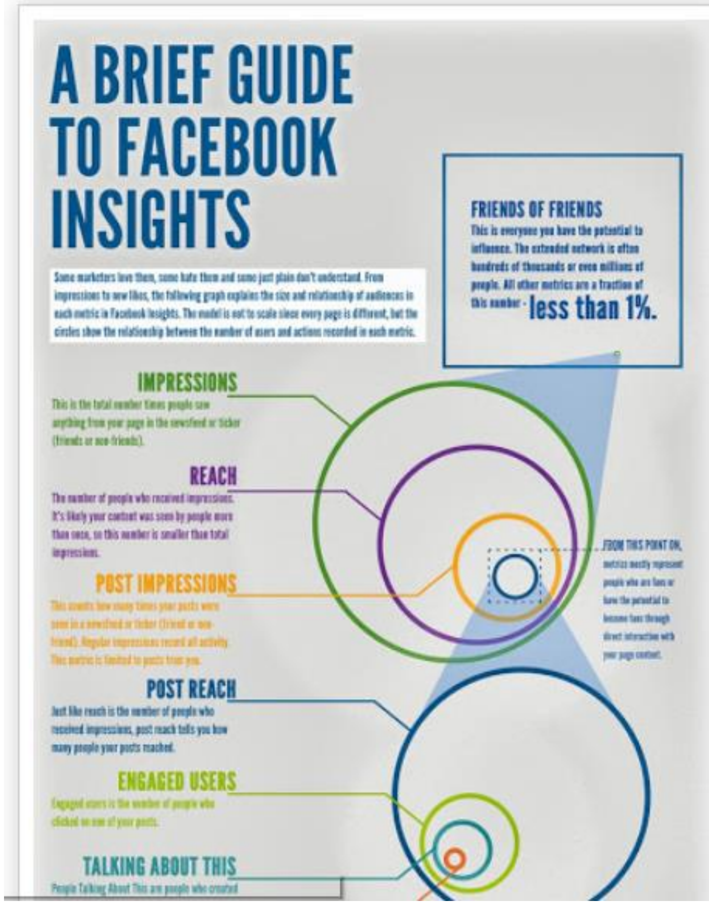
Facebook Launches Advanced AI Effort to Find Meaning in Your Posts

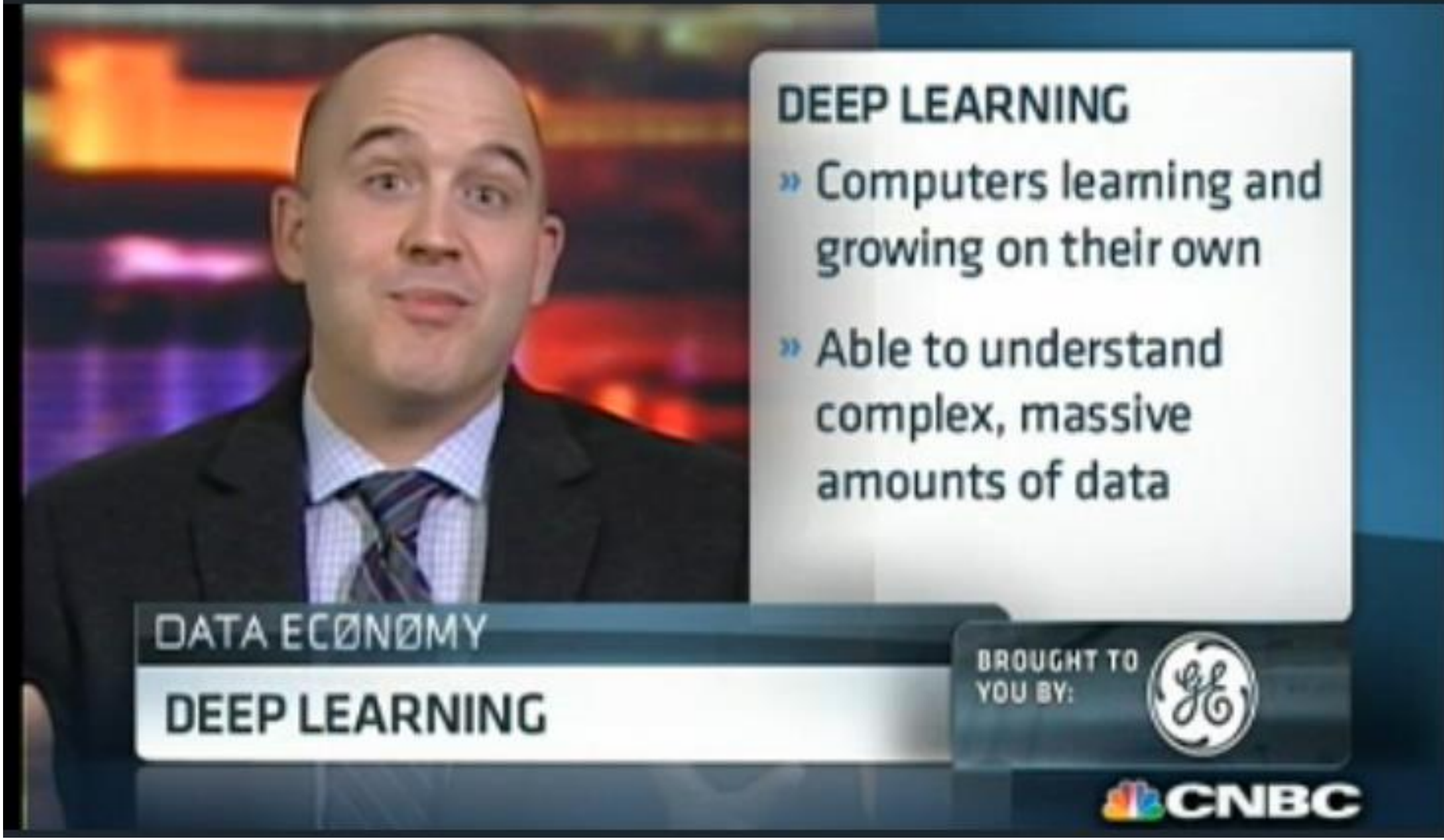
September 20, 2013

A technique called deep learning could help Facebook understand its users and their data better.

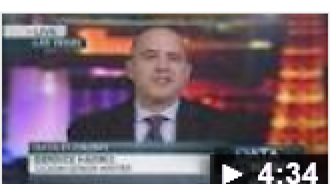
By Tom Simonite on September 20, 2013

……Facebook's foray into deep learning sees it following its competitors Google and Microsoft, which have used the approach to impressive effect in the past year. Google has hired and acquired leading talent in the field (see "[10 Breakthrough Technologies 2013: Deep Learning]"), and last year created software that taught itself to recognize cats and other objects by reviewing stills from YouTube videos. The underlying deep learning technology was later used to slash the error rate of Google's voice recognition services (see "[Google's Virtual Brain Goes to Work]")….Researchers at Microsoft have used deep learning to build a system that translates speech from English to Mandarin Chinese in real time (see "[Microsoft Brings Star Trek's Voice Translator to Life]"). Chinese Web giant Baidu also recently established a Silicon Valley research lab to work on deep learning.





[Is **Deep Learning**, the 'holy grail' of big data? - CNBC - Video]



video.cnbc.com/gallery/?video=3000192292

Aug 22, 2013

Derrick Harris, GigaOM, explains how "**Deep Learning**" computers are able to process and understand ...

MIT Technology Review

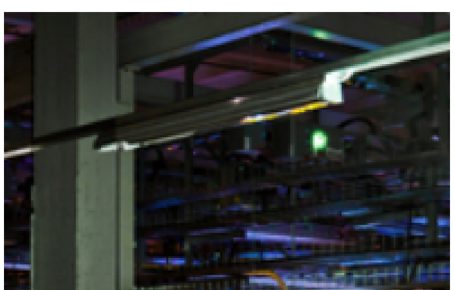
BUSINESS NEWS

8 COMMENTS

Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014



How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in

BloombergBusinessweek
Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. “DeepMind is bona fide in terms of its research capabilities and depth,” says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. “We would have more if the talent was there to be had,” he says. “Last year, the cost of a top, world-class deep learning expert was about the same as a top NFL quarterback prospect. The cost of that talent is pretty remarkable.”

ICASSP 2013

Vancouver Convention & Exhibition Centre

May 26 - 31, 2013 • Vancouver, Canada

IEEE

IEEE Signal Processing Society

- Home
- Photo Gallery
- Mobile App
- Organizing Committee
- **Plenary Speakers**
- Call for Papers
- Technical Program
- Registration
- Paper Submission
- SP Letters Presentation
- Special Sessions
- Hotels
- Tutorials
- Workshops
- Supporters
- Support And Exhibits
- Student Luncheon

Plenary Speakers

Geoffrey E. Hinton

University of Toronto and Google Inc.

View Video

Daphne Koller

Stanford University

- Photo Gallery
- Mobile App
- Organizing Committee
- Plenary Speakers
- Call for Papers
- Technical Program
- Registration
- Paper Submission
- SP Letters Presentation
- **Special Sessions**

Special Sessions

ICASSP 2013 will offer the following special sessions:

Acoustic Event Detection and Scene Analysis

Organized by Mark Plumbley, Dimitris Giannoulis and Mathieu Lagrange

New types of deep neural network learning for speech recognition and related applications

Organized by Li Deng, Geoff Hinton and Brian Kingsbury

Li Deng, Dong Yu, Geoffrey Hinton

Microsoft Research; Microsoft Research; University of Toronto

Deep Learning for Speech Recognition and Related Applications

7:30am - 6:30pm Saturday, December 12, 2009

Location: Hilton: Cheakamus

Abstract: Over the past 25 years or so, speech recognition technology has been dominated by a "shallow" architecture --- hidden Markov models (HMMs). Significant technological success has been achieved using complex and carefully engineered variants of HMMs. The next generation of the technology requires solutions to remaining technical challenges under diversified deployment environments. These challenges, not adequately addressed in the past, arise from the many types of variability present in the speech generation process. Overcoming these challenges is likely to require "deep" architectures with efficient learning algorithms. For speech recognition and related sequential pattern recognition applications, some attempts have been made in the past to develop computational architectures that are "deeper" than conventional HMMs, such as hierarchical HMMs, hierarchical point-process models, hidden dynamic models, and multi-level detection-based architectures, etc. While positive recognition results have been reported, there has been a conspicuous lack of systematic learning techniques and theoretical guidance to facilitate the development of these deep architectures. Further, there has been virtually no effective communication between machine learning researchers and speech recognition researchers who are both advocating the use of deep architecture and learning. One goal of the proposed workshop is to bring together these two groups of researchers to review the progress in both fields and to identify promising and synergistic research directions for potential future cross-fertilization and collaboration.

http://research.microsoft.com/en-us/um/people/dongyu/NIPS2009/

Microsoft Research

Useful Sites on Deep Learning

- http://www.cs.toronto.edu/~hinton/
- http://ufldl.stanford.edu/wiki/index.php/UFLDL_Recommended_Readings
- http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial (Andrew Ng's group)
- http://deeplearning.net/reading-list/ (Bengio's group)
- http://deeplearning.net/tutorial/
- **http://deeplearning.net/deep-learning-research-groups-and-labs/**
- Google+ Deep Learning community

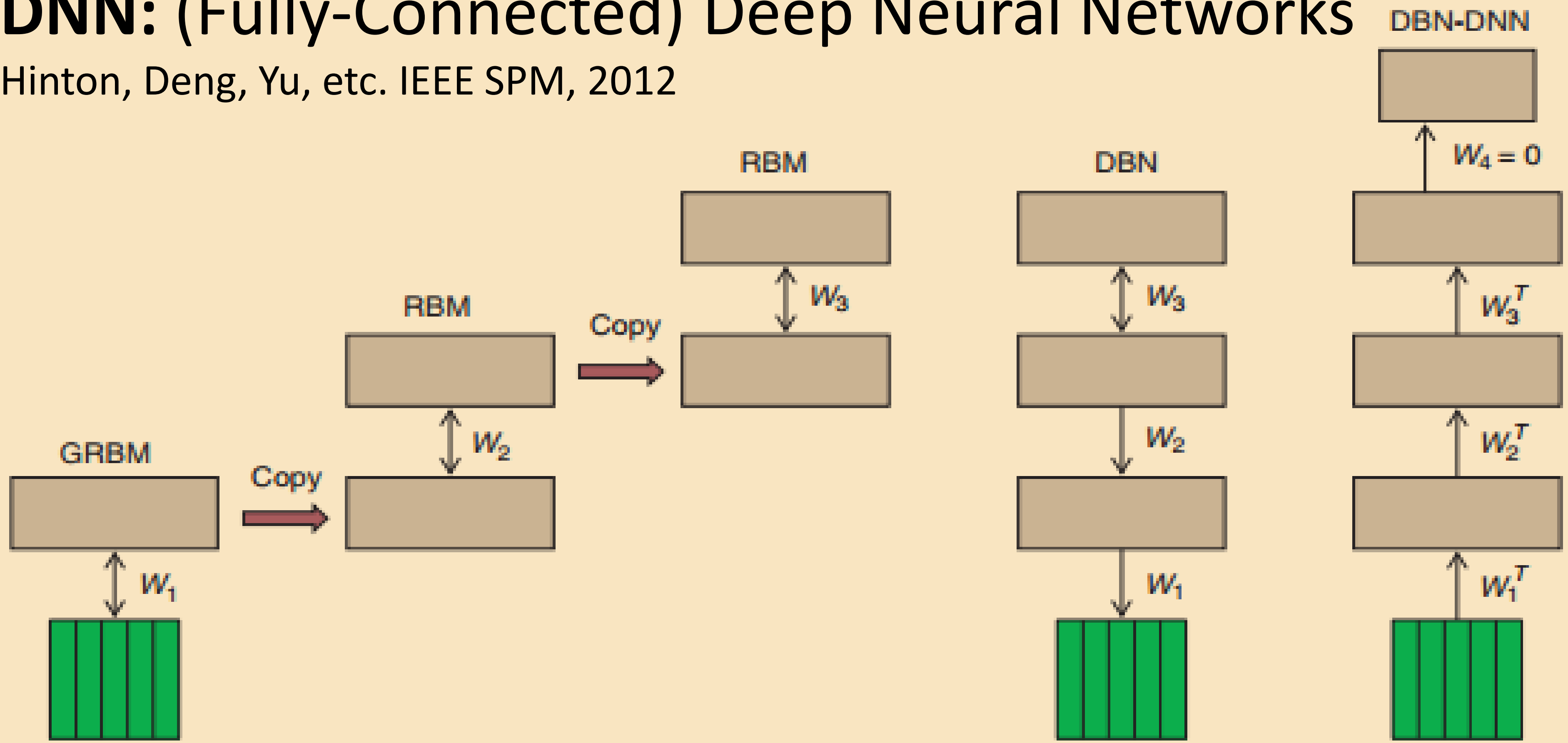
Part I

Background of Deep Learning

Common and NLP-centric architectures

DNN: (Fully-Connected) Deep Neural Networks

Hinton, Deng, Yu, etc. IEEE SPM, 2012



First train a stack of N models each of which has one hidden layer. Each model in the stack treats the hidden variables of the previous model as data.

Then compose them into a single Deep Belief Network.

Then add outputs and train the DNN with backprop.

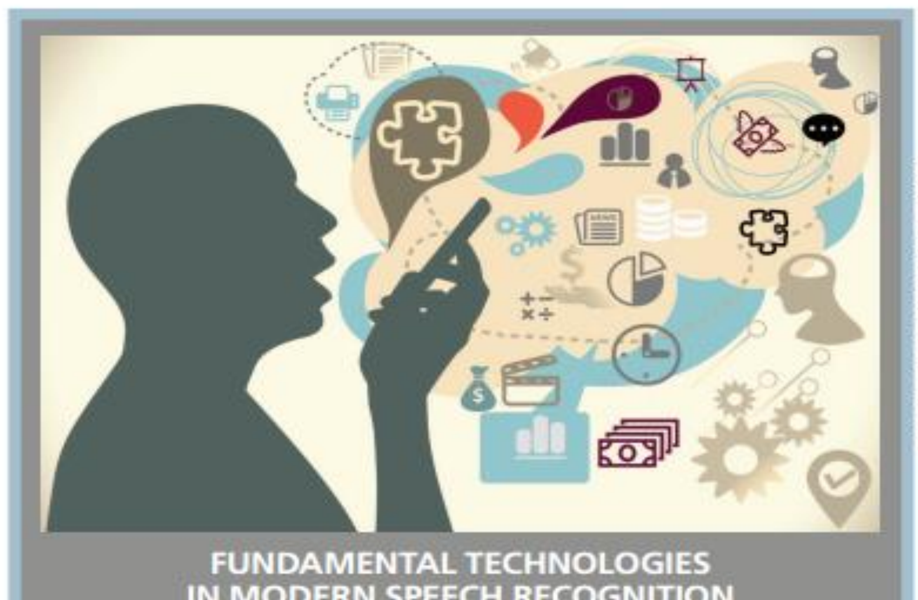
Microsoft Research

IEEE Signal Processing Magazine,
vol. 29, no. 6, pp. 82-97, Nov. 2012

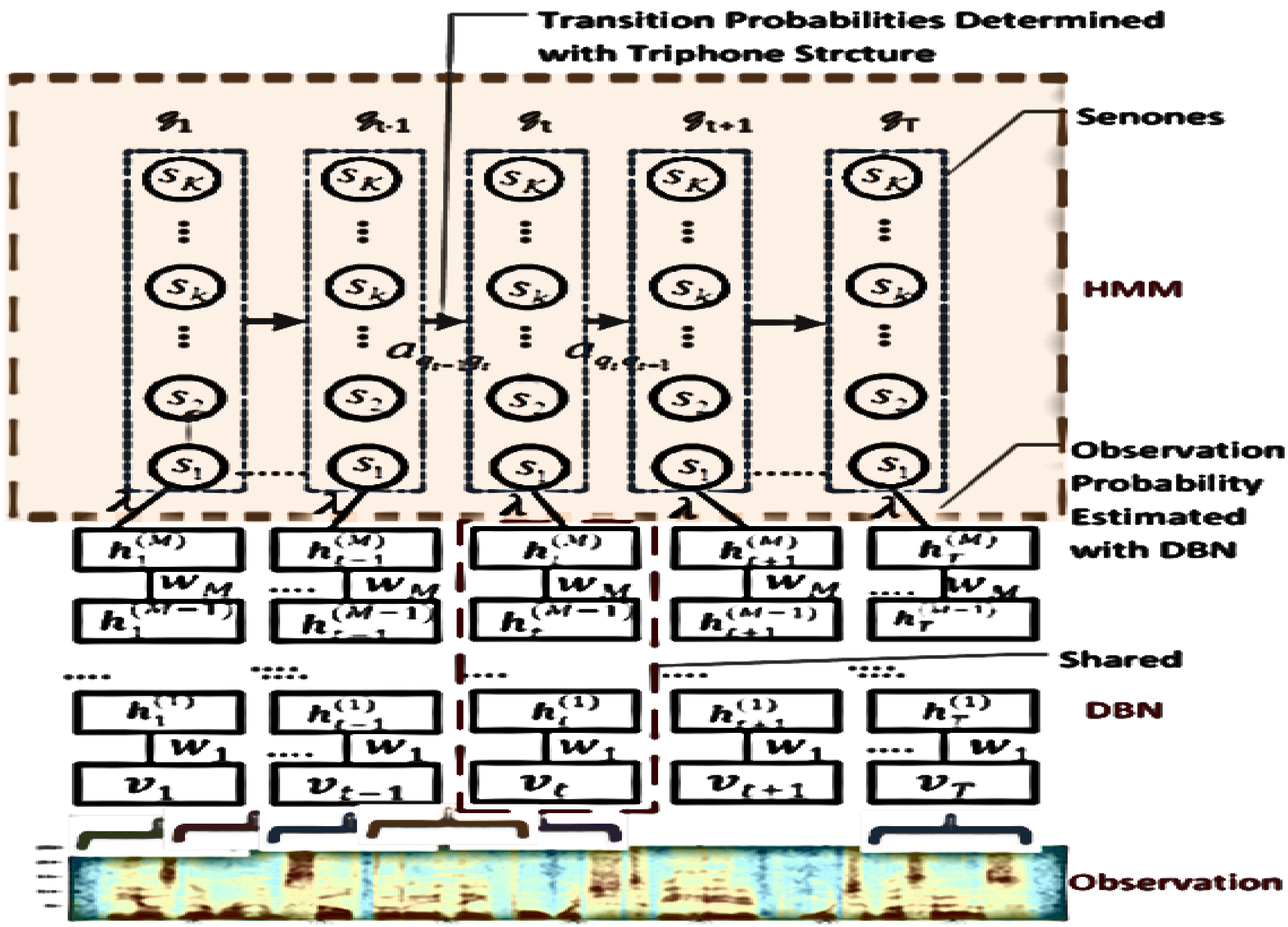
Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury

Deep Neural Networks for Acoustic Modeling in Speech Recognition

The shared views of four research groups



Context-Dependent DNN-HMM (2010 at MSR for speech recognition)



Microsoft Research

DNN-HMM vs. GMM-HMM

(Deng, Yu, Acero, 2006; Mohamed, Yu, Deng, 2010; Yu, Deng, Dahl, 2010-2012; Seide, Li, Yu 2011; Chen,Li,Seide,Yu,2012)

- **Table:** TIMIT Phone recognition (3 hours of training)

Features	Setup	Error Rates
GMM	w. deep hid.dynamics	24.8%
DNN	5 layers x 2048	22.8%

- **Table:** Voice Search SER (24-48 hours of training)

Features	Setup	Error Rates
GMM	MPE (760 24-mix)	36.2%
DNN	5 layers x 2048	30.1%

- **Table:** Switch Board WER (309 hours training)

Features	Setup	Error Rates
GMM	BMMI (9K 40-mix)	23.6%
DNN	7 layers x 2048	15.8%

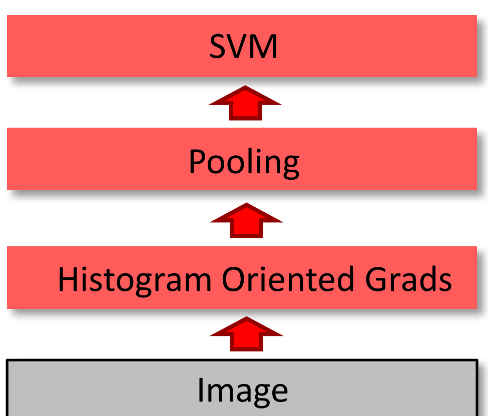
- **Table:** Switch Board WER (2000 hours training)

Features	Setup	Error Rates
GMM	BMMI (18K 72-mix)	21.7%
DNN	7 layers x 2048	14.6%

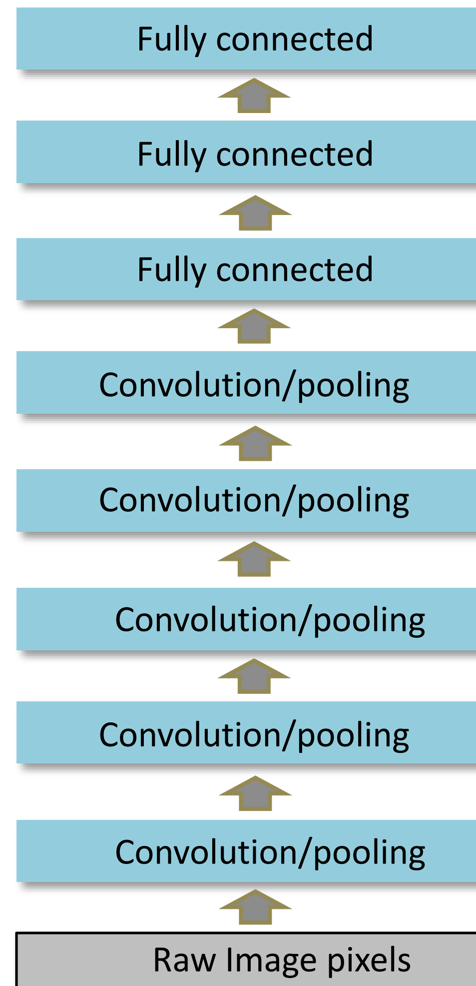
Deep **Convolutional NN** for Images

CNN: local connections with weight sharing; pooling for translation invariance

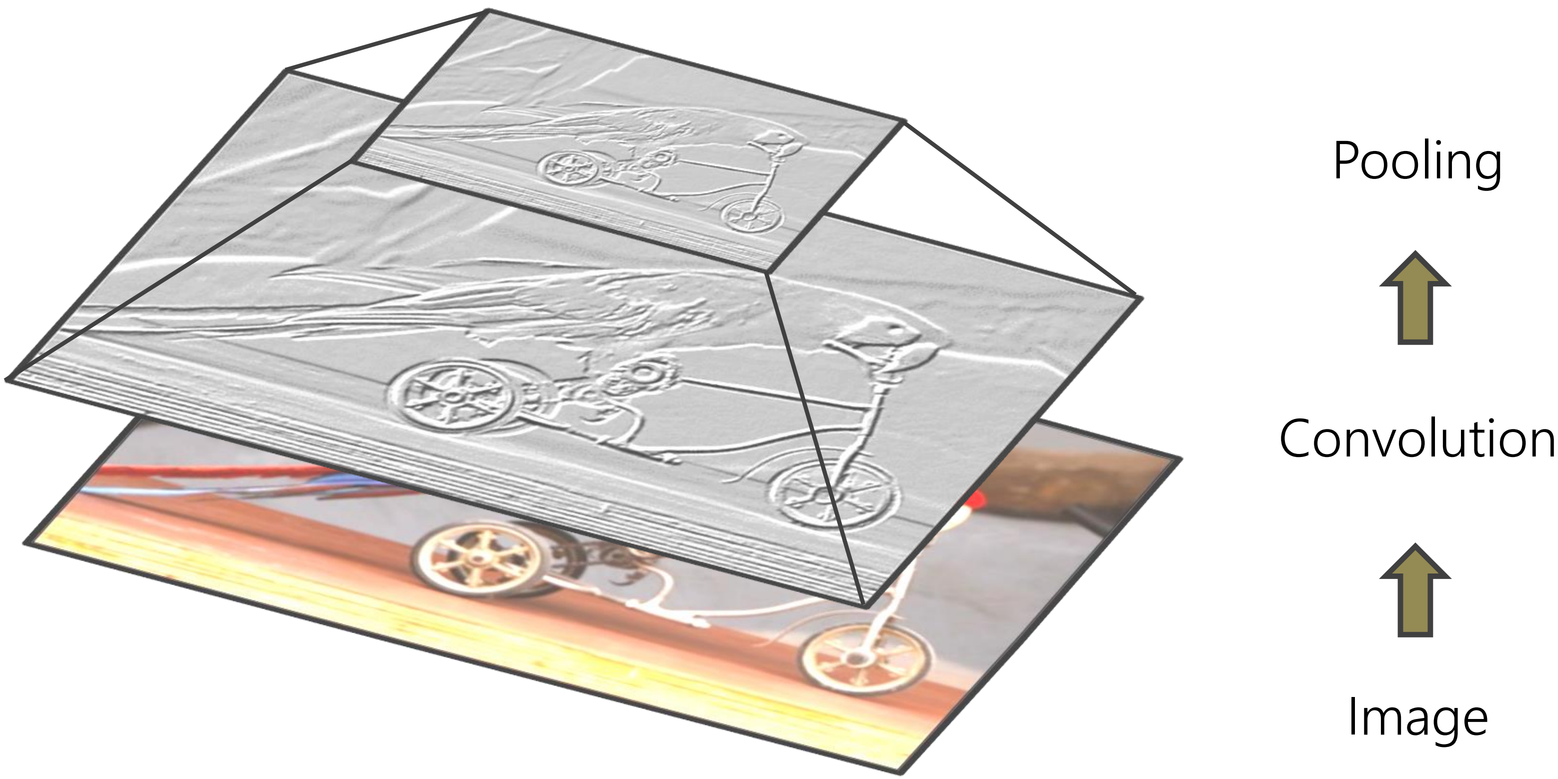
earlier



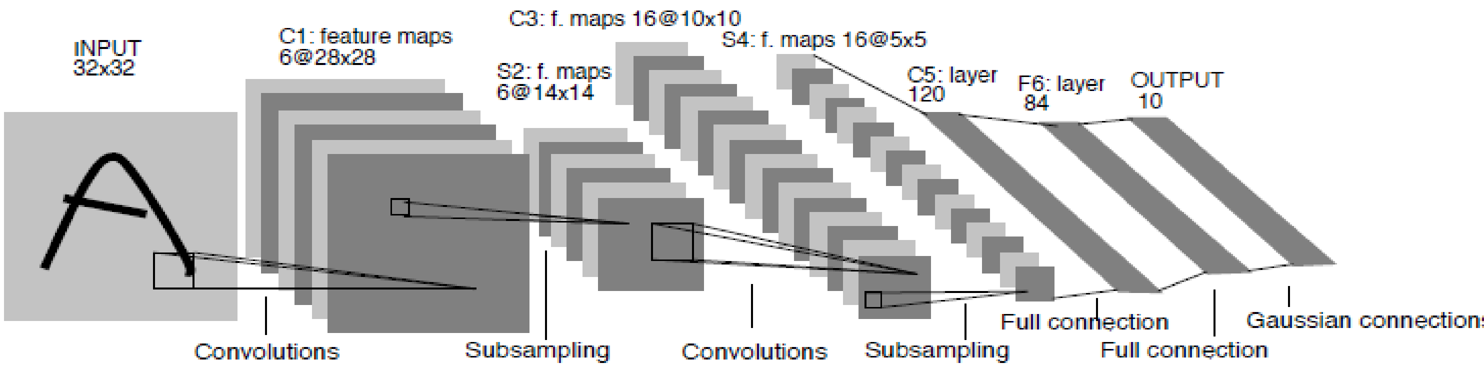
2012-2013



A Basic Module of the CNN

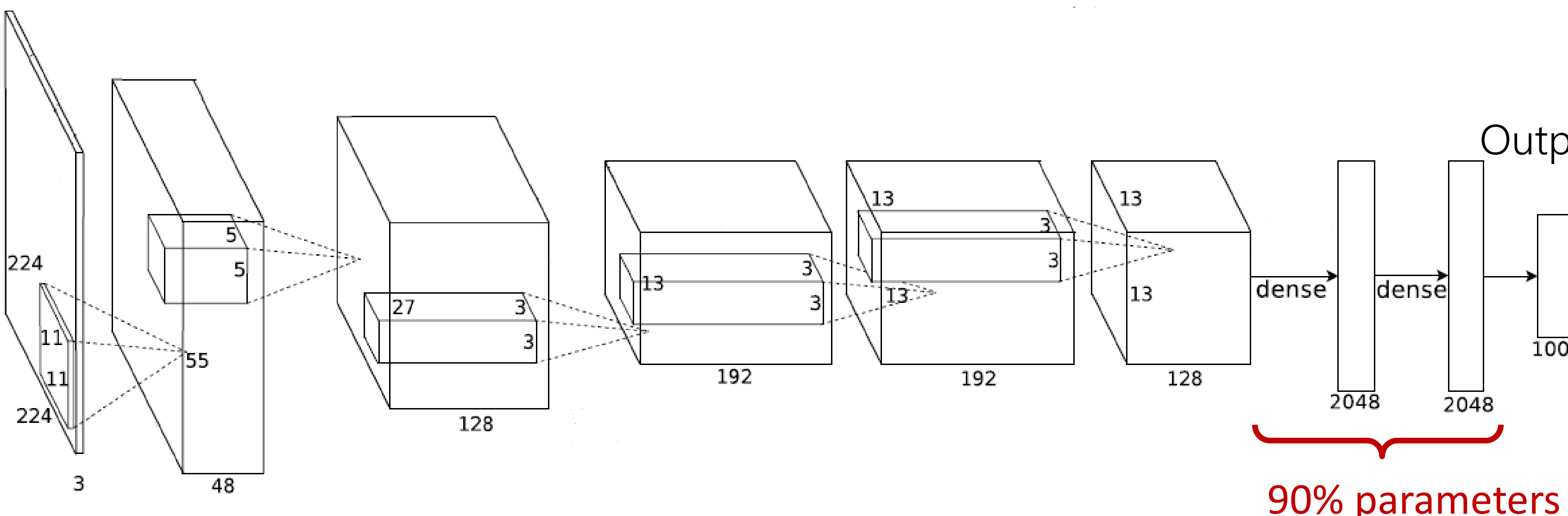


Deep CNN



Image

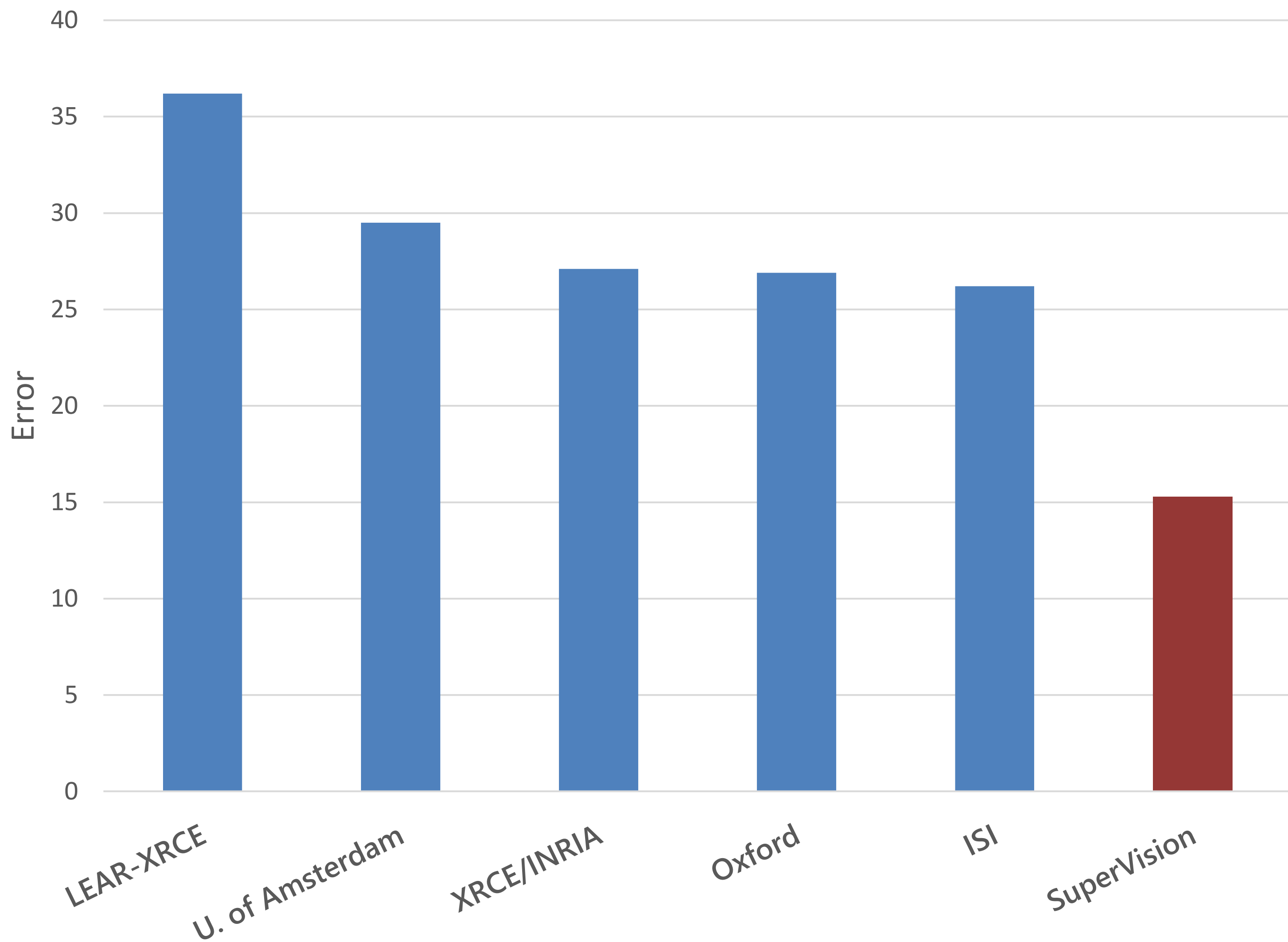
LeCun et al., 1998



SuperVision, 2012

ImageNet 1K Competition

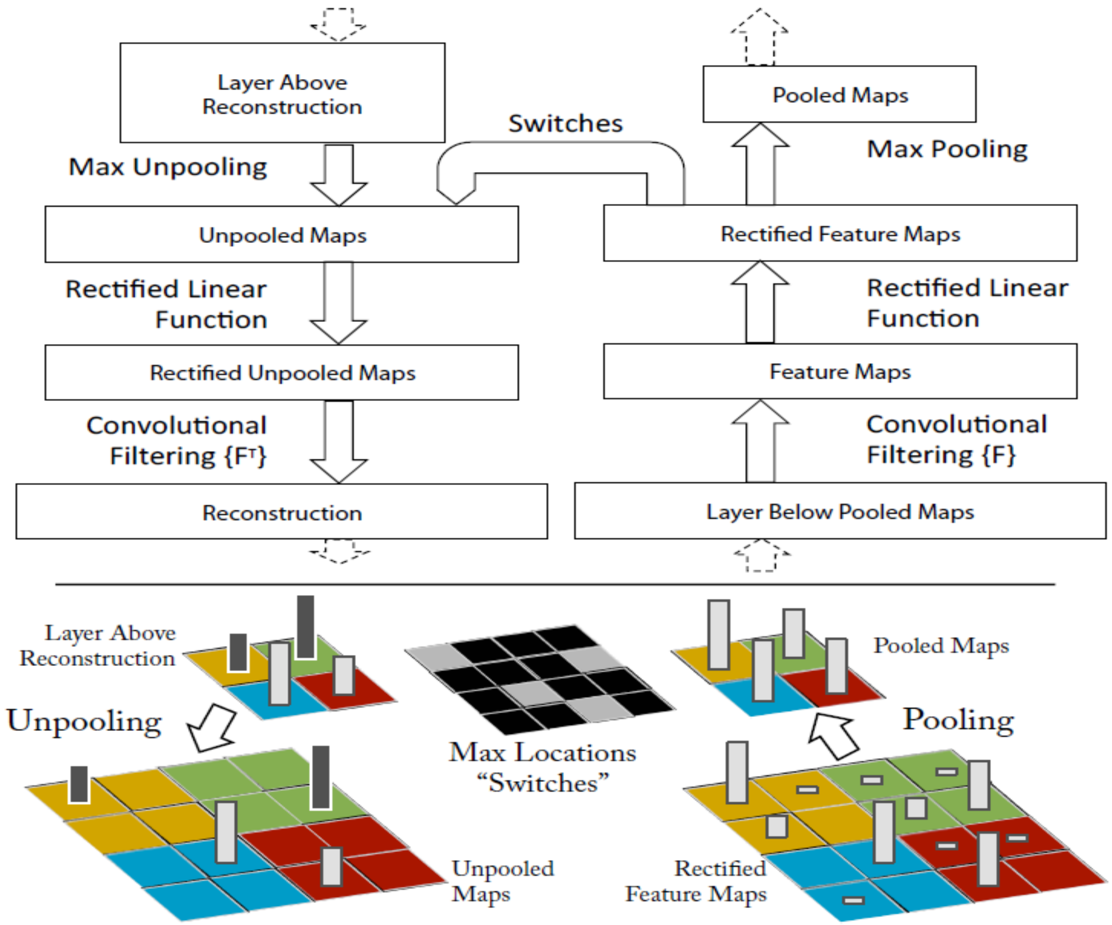
(Fall 2012)



Deep CNN !!!
Univ. Toronto team

Microsoft Research

Deconvolutional Neural Nets

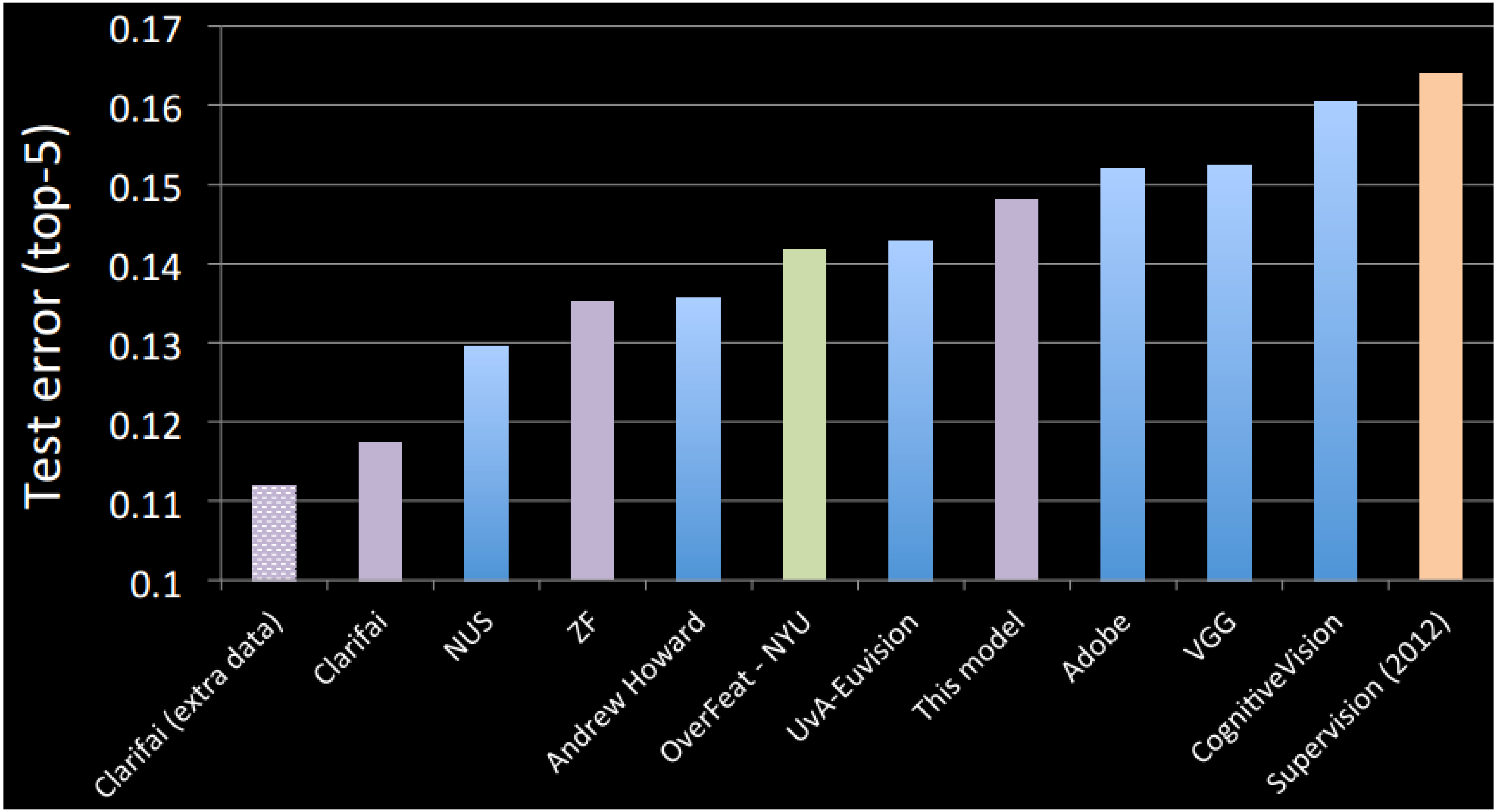


The top portion shows how a deconvolutional network's layer (left) is attached to a corresponding CNN's layer (right). The deconvolutional network reconstructs an approximate version of the CNN features from the layer below. The bottom portion is an illustration of the unpooling operation in the deconvolutional network, where "Switches" are used to record the location of the local max in each pooling region during pooling in the CNN. [after (Zeiler and Fergus, 2013), @arXiv].

Microsoft Research

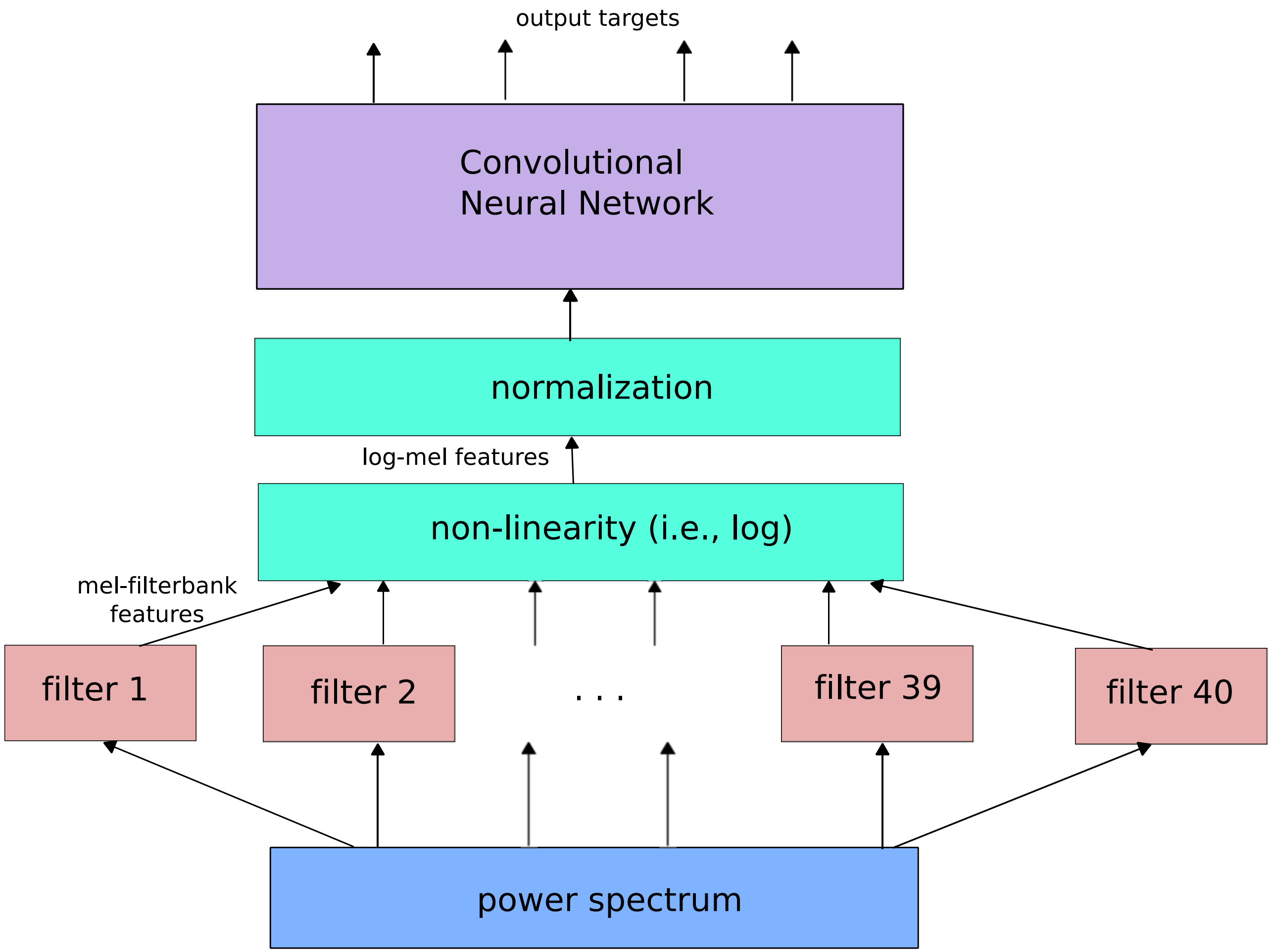
Same ImageNet 1K Competition

One year later (Fall 2013)



Summary results of ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013), representing the state-of-the-art performance of object recognition systems.

CNN also good for speech

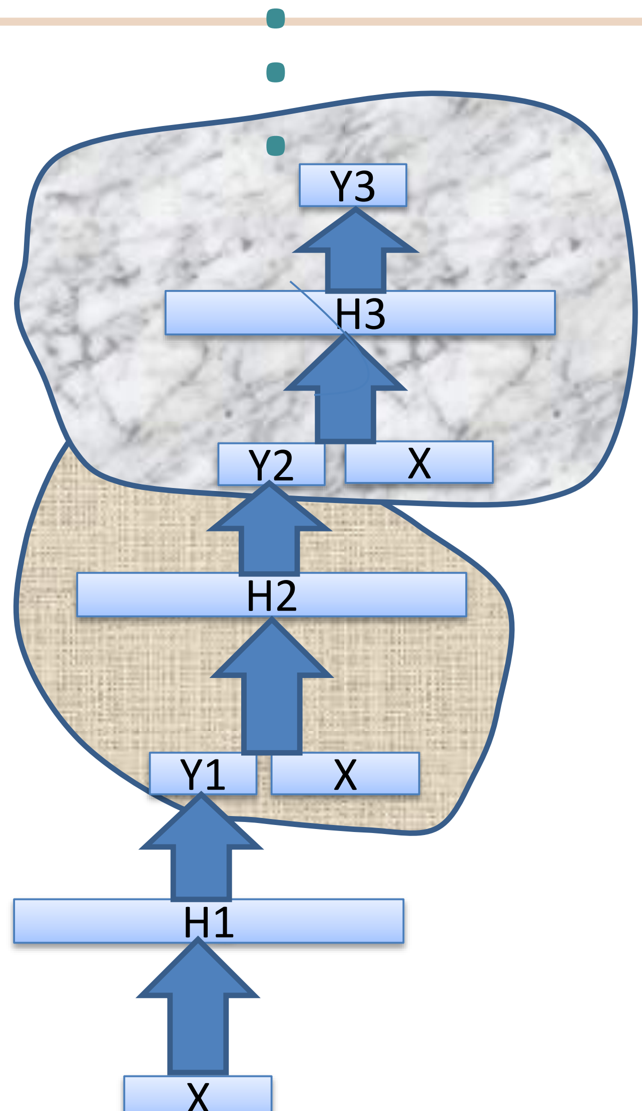


Joint learning of filter parameters and the rest of the deep network. [after (Sainath et al., 2013b), @IEEE].

Microsoft Research

Deep Stacking Network (DSN)

- Interleave linear/nonlinear layers
- Exploit closed-form constraints among network's weights
- Much easier to learn than DNN
- Naturally amenable to parallel training
- (Largely) convex optimization

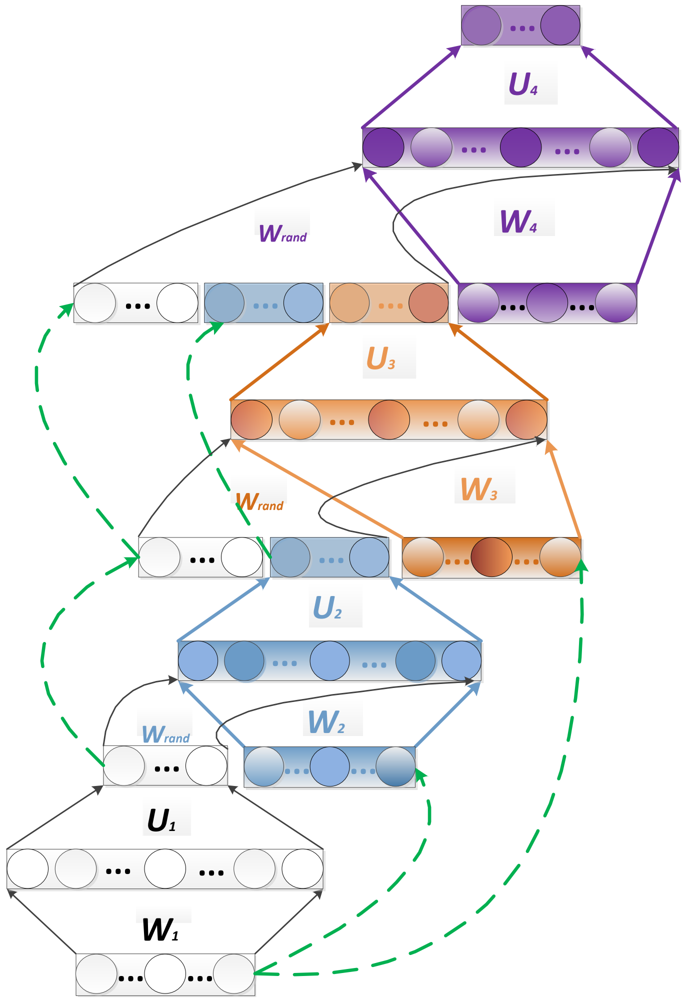


Deng, Yu, Platt: [Scalable stacking and learning for building deep architectures](), IEEE *ICASSP* , March 2012

Microsoft Research

Learning DSN Weights --- Main Ideas

- Learn weight matrices U and W in individual modules separately.
- Given W and linear output layer, U can be expressed as explicit nonlinear function of W.
- This nonlinear function is used as the constraint in solving nonlinear least square for learning W.
- Initializing W with RBM (bottom layer)
- For higher layers, part of W is initialized with the optimized W from the immediately lower layer and part of it with random numbers

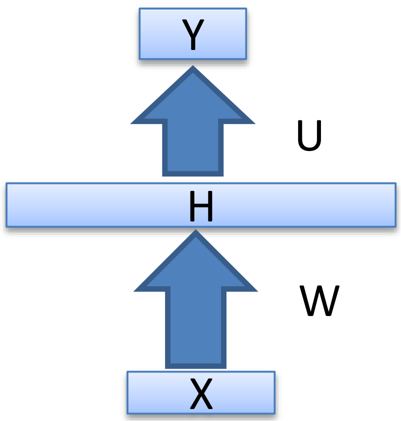


A neat way of learning DSN weights

$$\mathrm{E} = \frac{1}{2}\sum_n ||\boldsymbol{y}_n - \boldsymbol{t}_n||^2, \qquad \text{where } \boldsymbol{y}_n = \boldsymbol{U}^T\boldsymbol{h}_n = \boldsymbol{U}^T\sigma(\boldsymbol{W}^T\boldsymbol{x}_n) = G_n(\boldsymbol{U},\boldsymbol{W})$$

$$\frac{\partial E}{\partial \boldsymbol{U}} = 2\boldsymbol{H}(\boldsymbol{U}^T\boldsymbol{H} - \boldsymbol{T})^T \rightarrow \quad \boldsymbol{U} = \left(\boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{H}\boldsymbol{T}^T = \mathrm{F}(\boldsymbol{W}), \text{ where } \boldsymbol{h}_n = \sigma(\boldsymbol{W}^T\boldsymbol{x}_n)$$

$\mathrm{E} = \frac{1}{2}\sum_n ||G_n(\boldsymbol{U},\boldsymbol{W}) - \boldsymbol{t}_n||^2$, *subject to* $\boldsymbol{U} = \mathrm{F}(\boldsymbol{W})$,



Use of Lagrange multiplier method:

$$\mathrm{E} = \frac{1}{2}\sum_n ||G_n(\boldsymbol{U},\boldsymbol{W}) - \boldsymbol{t}_n||^2 + \lambda\, ||\boldsymbol{U} - \mathrm{F}(\boldsymbol{W})||$$

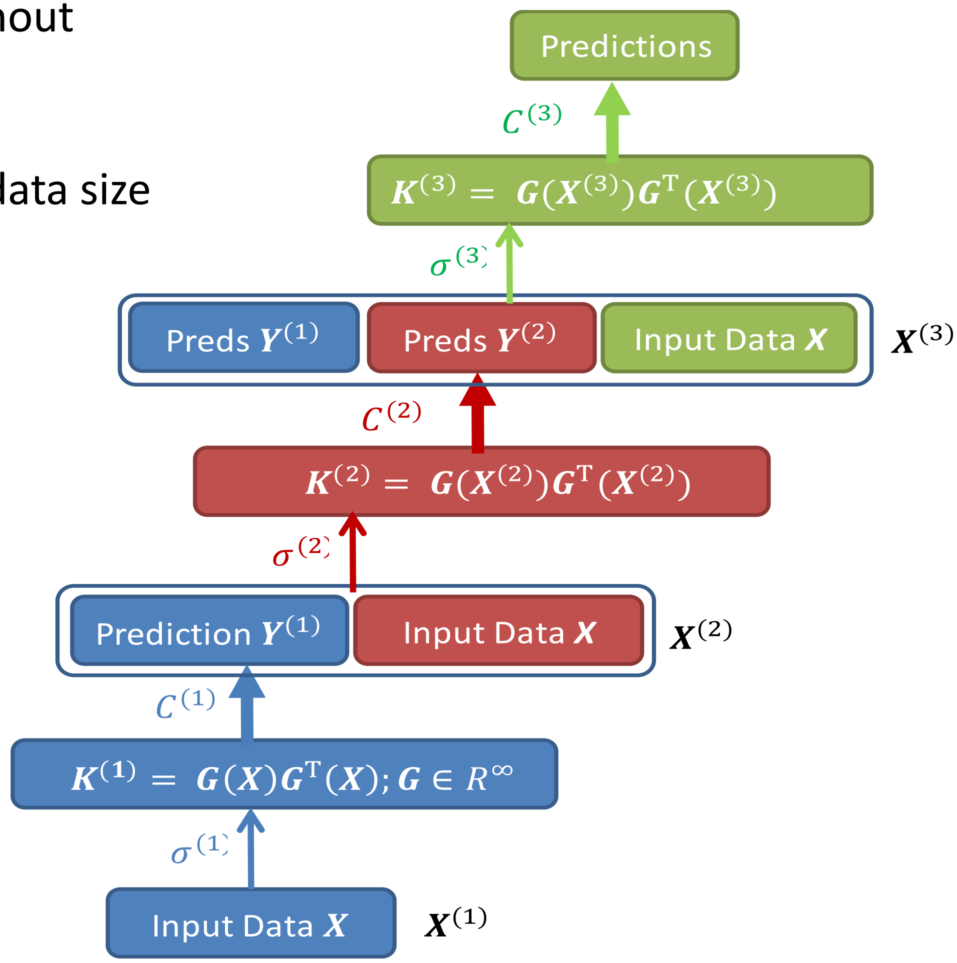
to learn ***W*** and then ***U*** → no longer backpropagation

- Advantages found:
  --- less noise in gradient than using chain rule ignoring explicit constraint $\boldsymbol{U} = \mathrm{F}(\boldsymbol{W})$
  --- batch learning is effective, aiding parallel training

Microsoft Research

Kernelized DSN: equivalent of inf-sized hidden layers

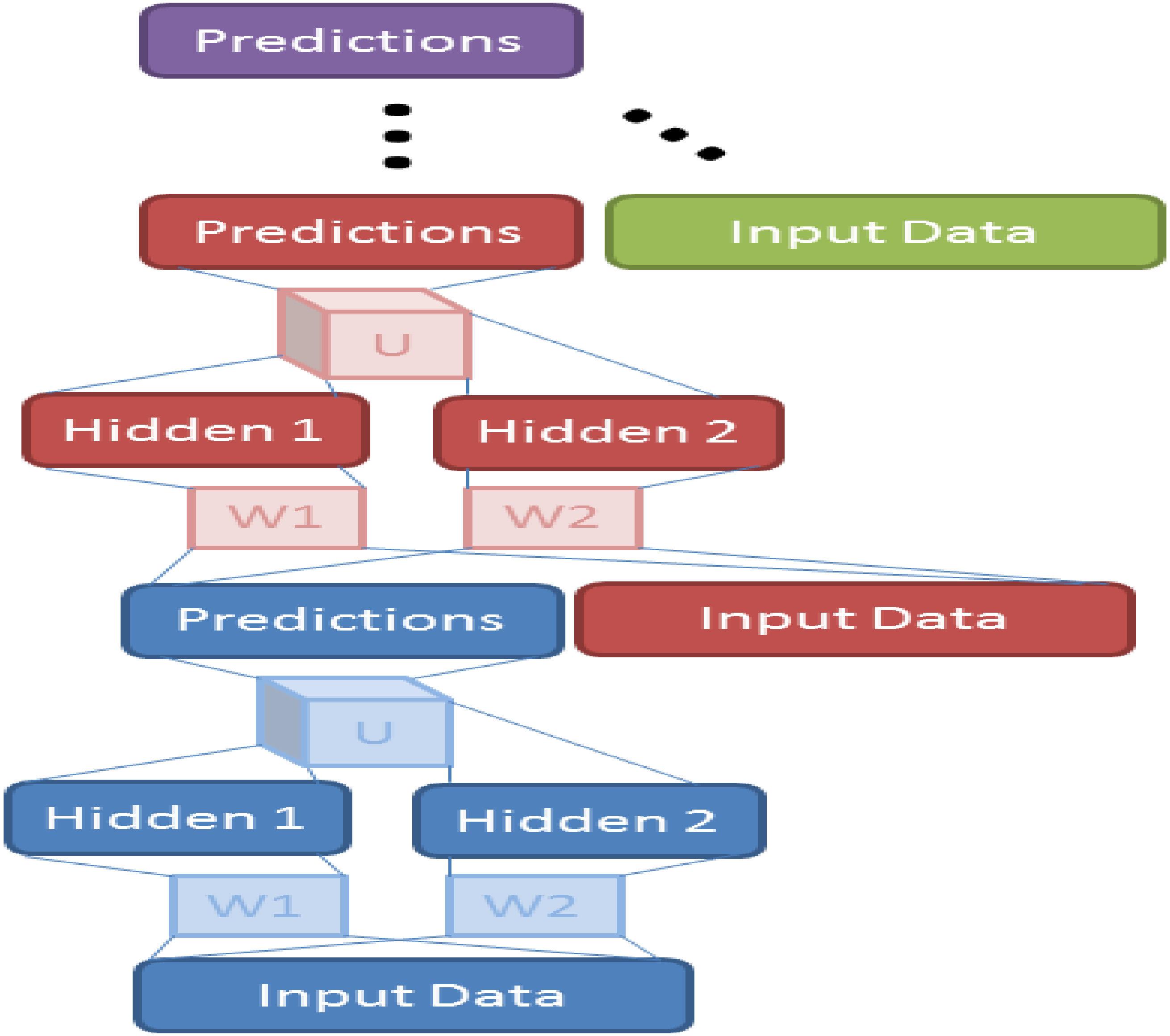
- Getting infinite-sized hidden layers without infinite-sized parameters
- Kernel trick is used
- Problem of kernel machine: Scaling to data size
- Lots of work done on approximation



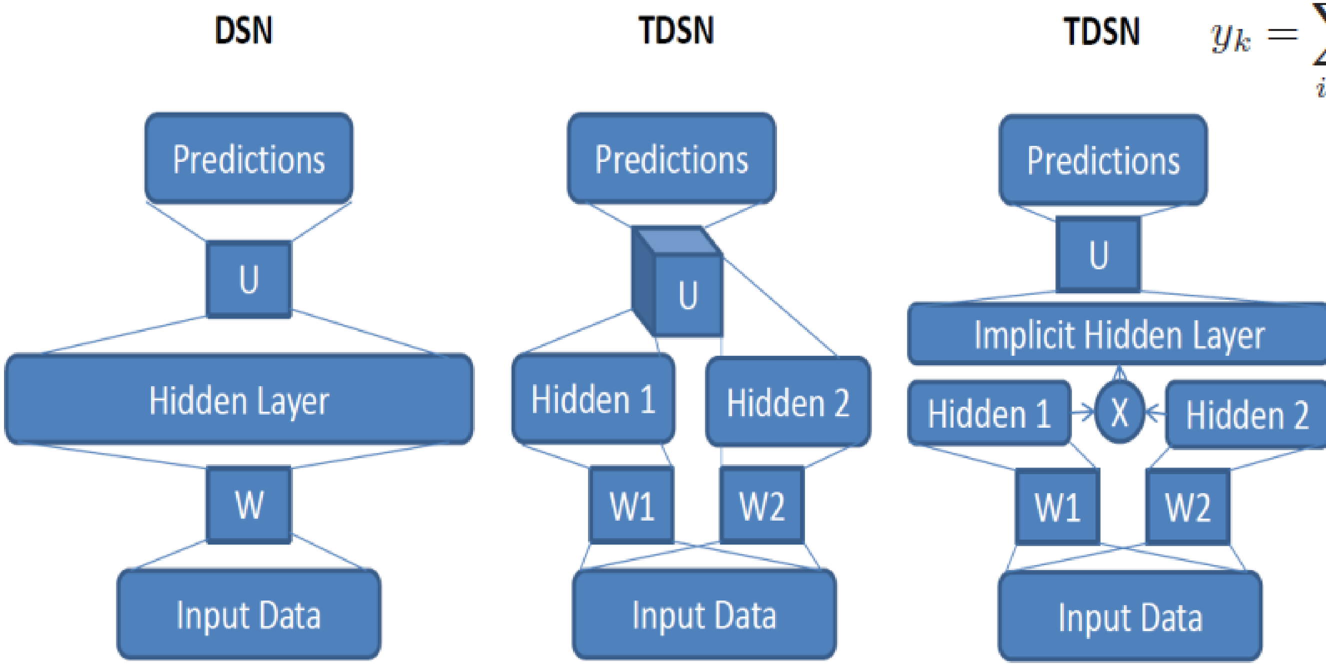
An example architecture of the K-DSN with three modules each of which uses a Gaussian kernel with different kernel parameters. [after (Deng, Tur, He, 2012), @IEEE]

Tensor Version of the DSN

(Hutchinson, Deng, & Yu, ICASSP-2012, IEEE T-PAMI, 2013)



Tensor-DSN is powerful: Correlation modeling of internal representations



$$y_k = \sum_{i=1}^{L_1} \sum_{j=1}^{L_2} \mathcal{U}_{ijk} h_{(1)i} h_{(2)j} = \tilde{\mathbf{u}}_k^T \tilde{\mathbf{h}}.$$

$$\tilde{\mathbf{u}}_k = \mathrm{vec}(\mathbf{U}_k) \in \mathbb{R}^{L_1 L_2}$$

$$\tilde{\mathbf{h}} = \mathbf{h}_{(1)} \otimes \mathbf{h}_{(2)} \in \mathbb{R}^{L_1 L_2}$$

Comparisons of a single module of a DSN (left) and that of a tensor DSN (TDSN). Two equivalent forms of a TDSN module are shown to the right. [after (Hutchinson et. al., 2012, 2013), @IEEE]

Tensor Version of the DNN

(Yu, Deng, Seide, IEEE T-ASLP, 2013)

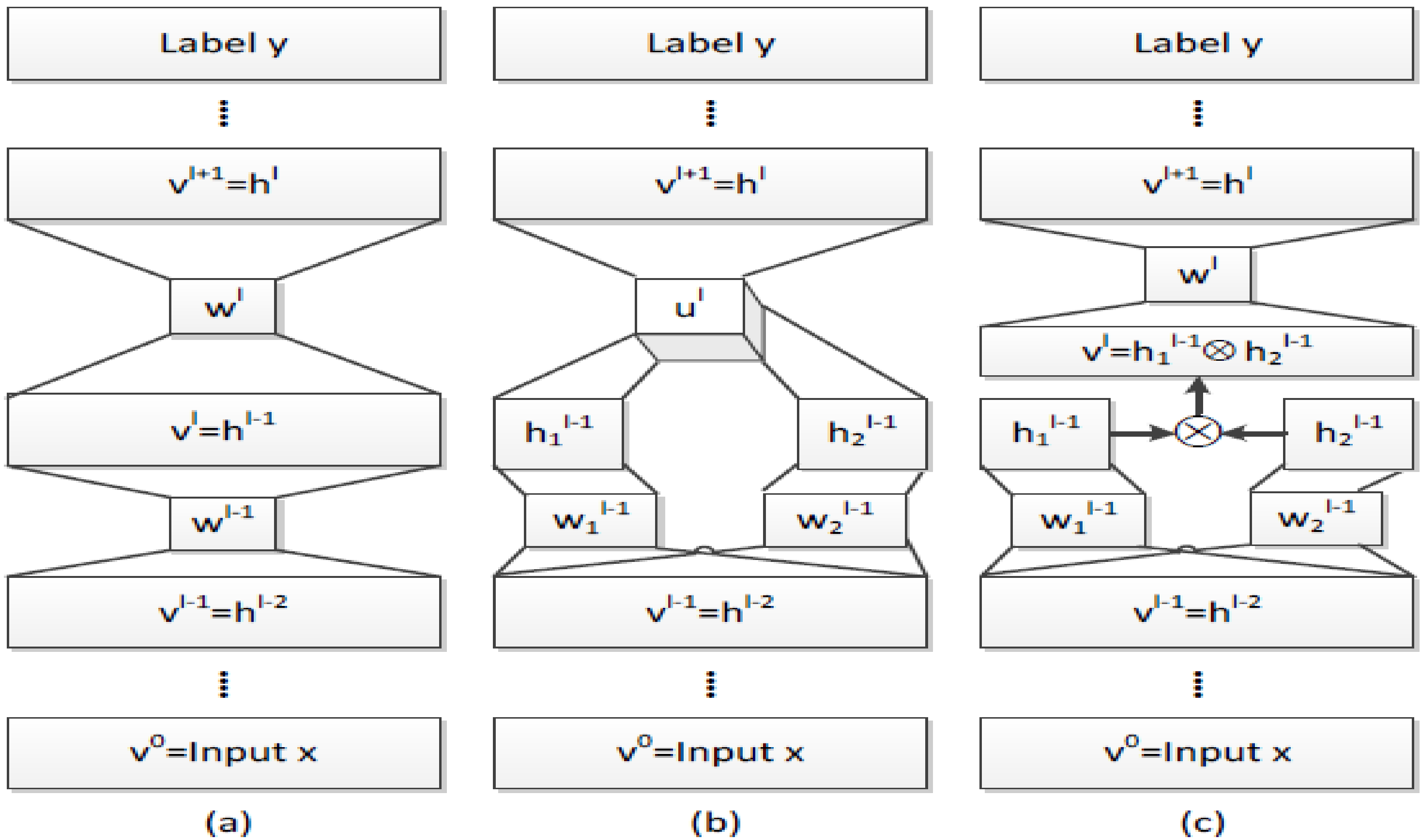
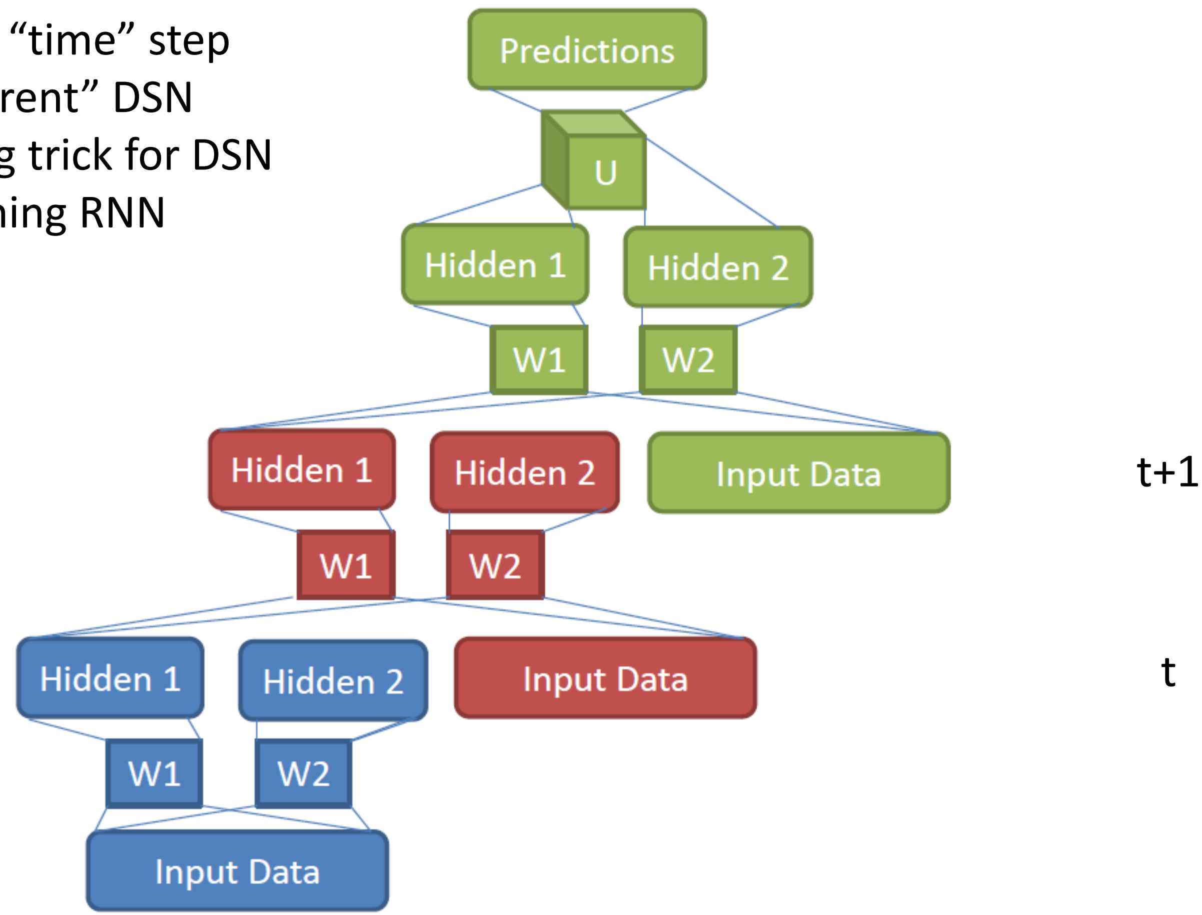


Figure 1: *Architectural illustrations of DNN and DTNN. (a) DNN. (b) DTNN: hidden layer h^{l-1} consists of two parts: h_1^{l-1} and h_2^{l-1}. Hidden layer h^l is a tensor layer to which the connection weights u^l form a three-way tensor. (c) An alternative representation of (b): tensor u^l is replaced with matrix w^l when v^l is defined as the cross product $h_1^{l-1} \otimes h_2^{l-1}$.*

Stacking with (double-small) Hidden Layers

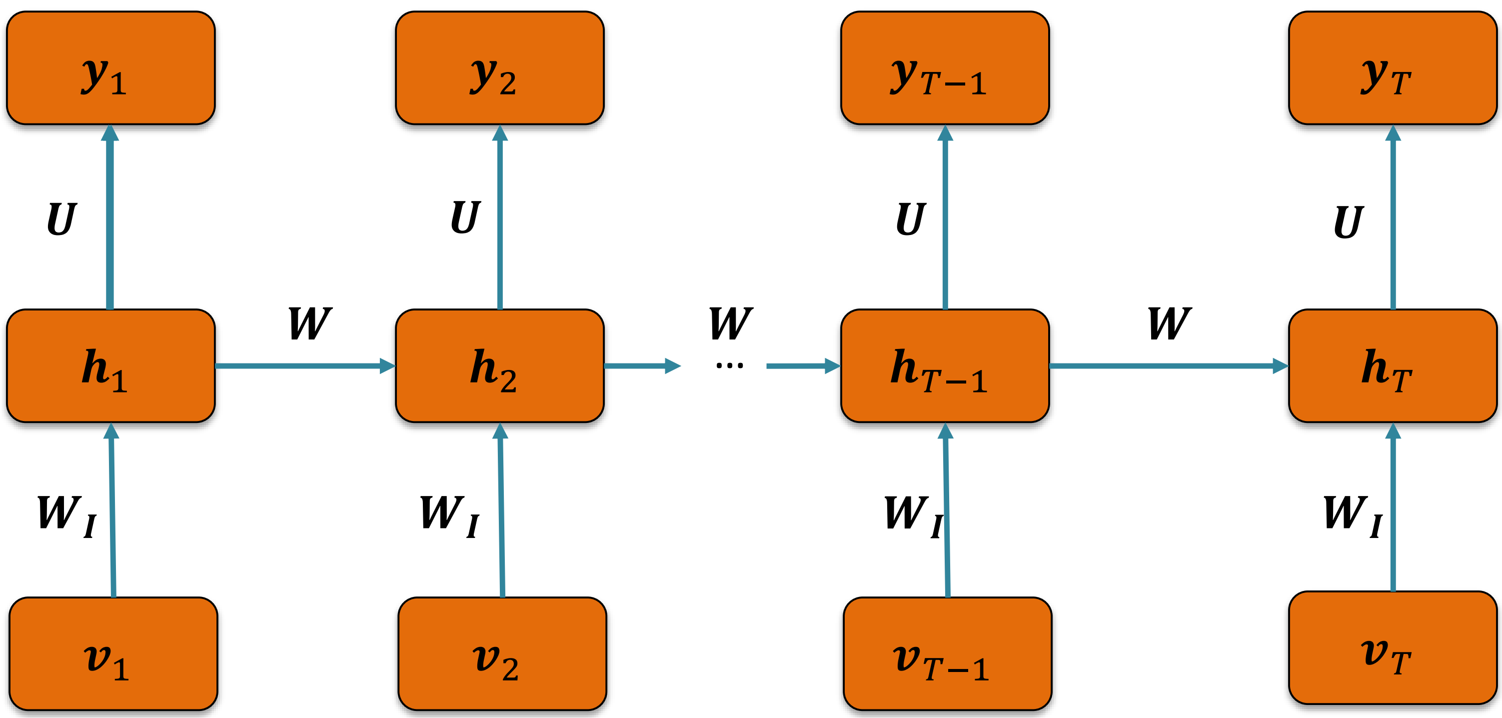
- Smaller-sized hidden stacking layer
- Closer to "Recurrent" neural nets
- A new module is a new "time" step
- This leads to the "recurrent" DSN
- Then, the same learning trick for DSN applies directly to learning RNN



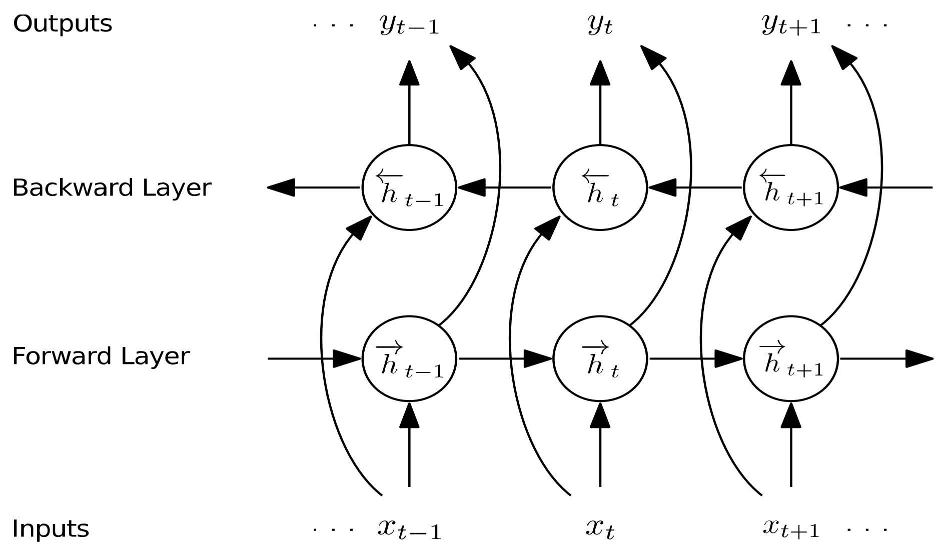
Stacking of TDSN modules by concatenating two hidden-layers' vectors with the input vector.

Recurrent Neural Networks (RNN)

- Recurrent Neural Network unfolding over time:



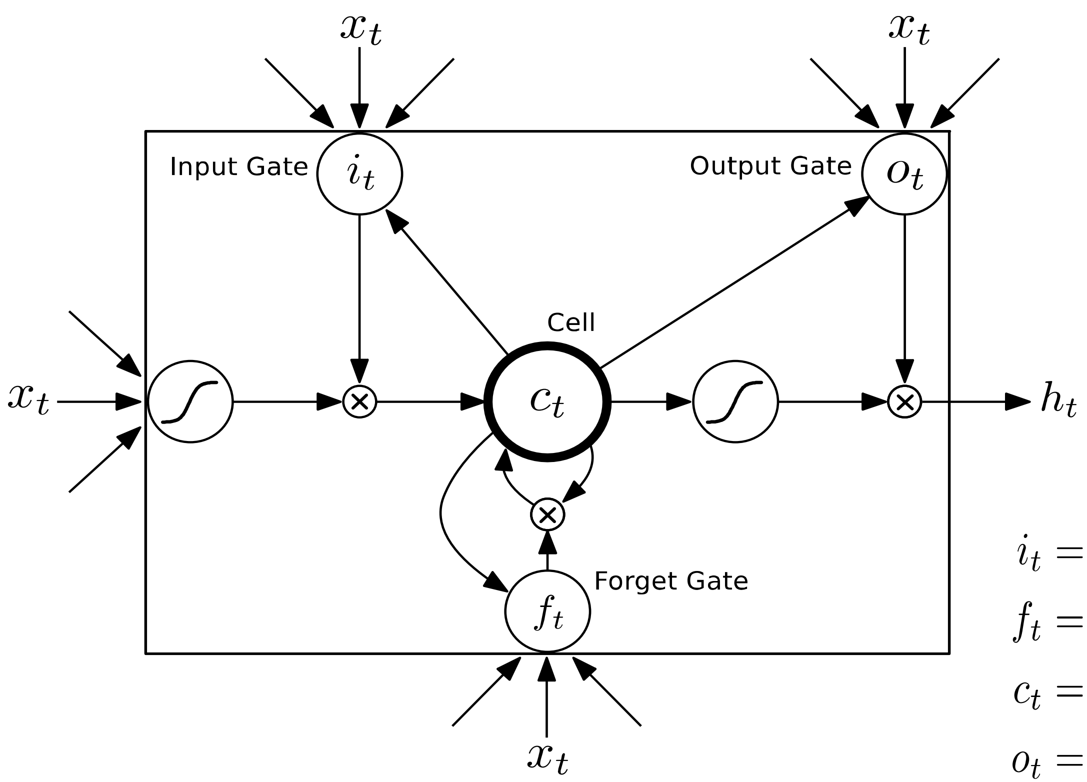
Bi-directional RNN



$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$
$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$
$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

Information flow in the bi-directional RNN, with both diagrammatic and mathematical descriptions. W's are weight matrices, not shown but can be easily inferred in the diagram. [after (Graves et al., 2013), @IEEE].

A Long-Short-Term-Memory Unit in LSTM-RNN



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Information flow in an LSTM unit of the RNN, with both diagrammatic and mathematical descriptions. W's are weight matrices, not shown but can easily be inferred in the diagram. [after (Graves et al., 2013), @IEEE].

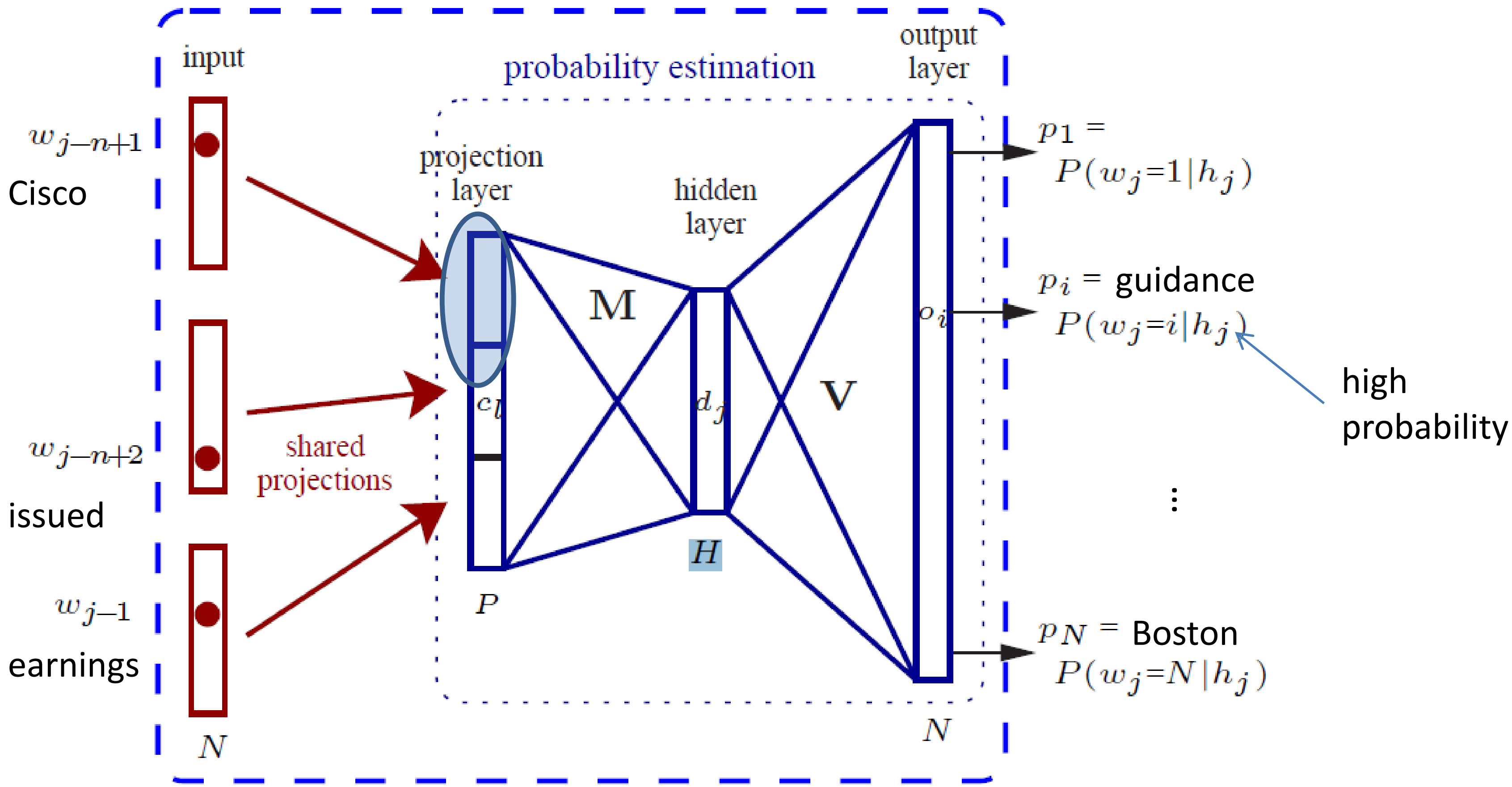
Microsoft Research

Part I

Background of Deep Learning

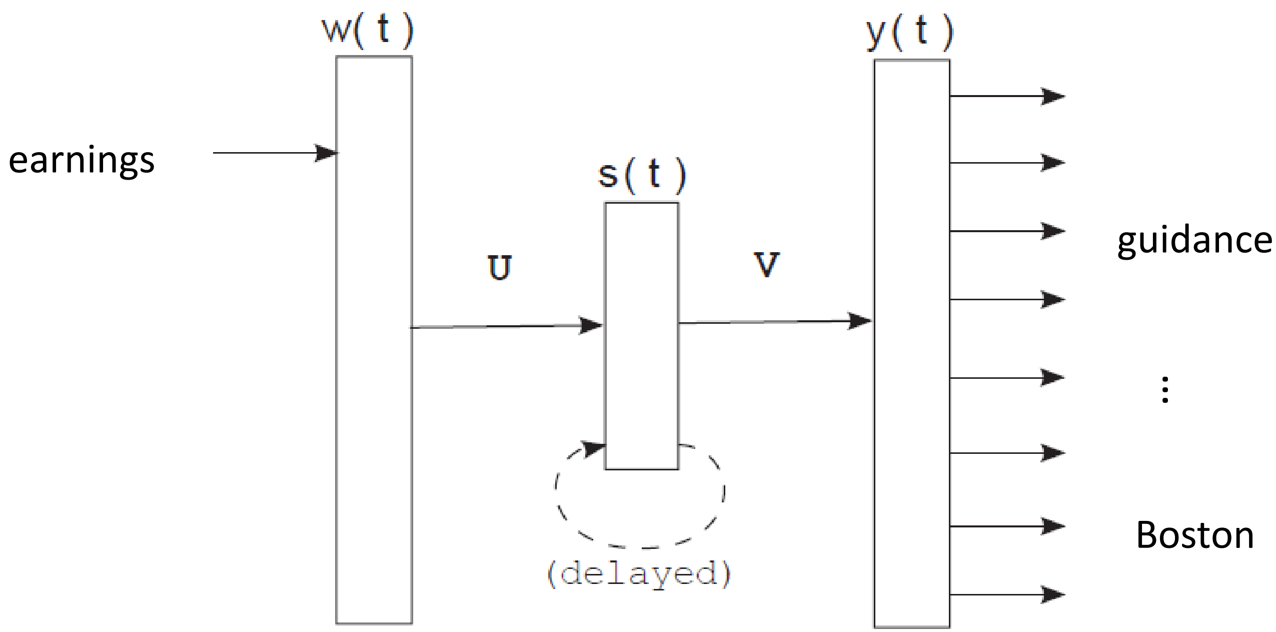
Common and NLP-Centric architectures

Neural-Network Language Models



Bengio et al., 2003;
Schwenk et al., 2006

Recurrent Neural Network for Language Modeling

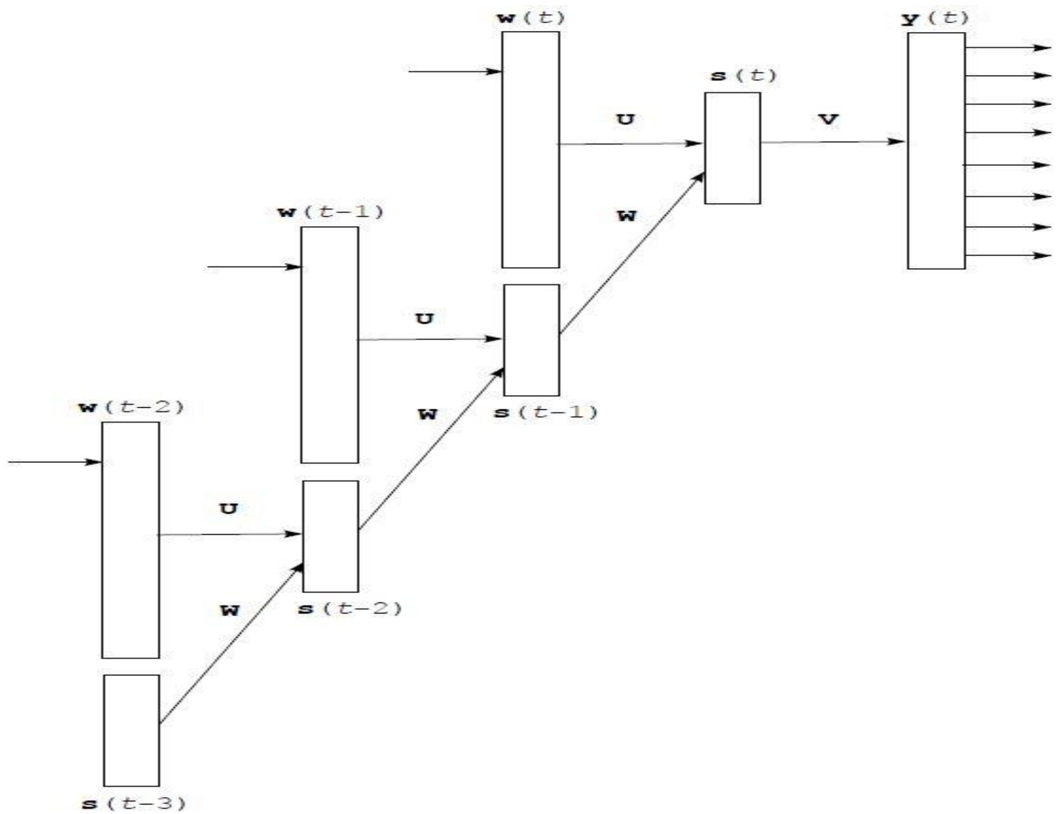


RNN::FFNN <----> IIR-Filter::FIR-Filter

Mikolov et al., 2011

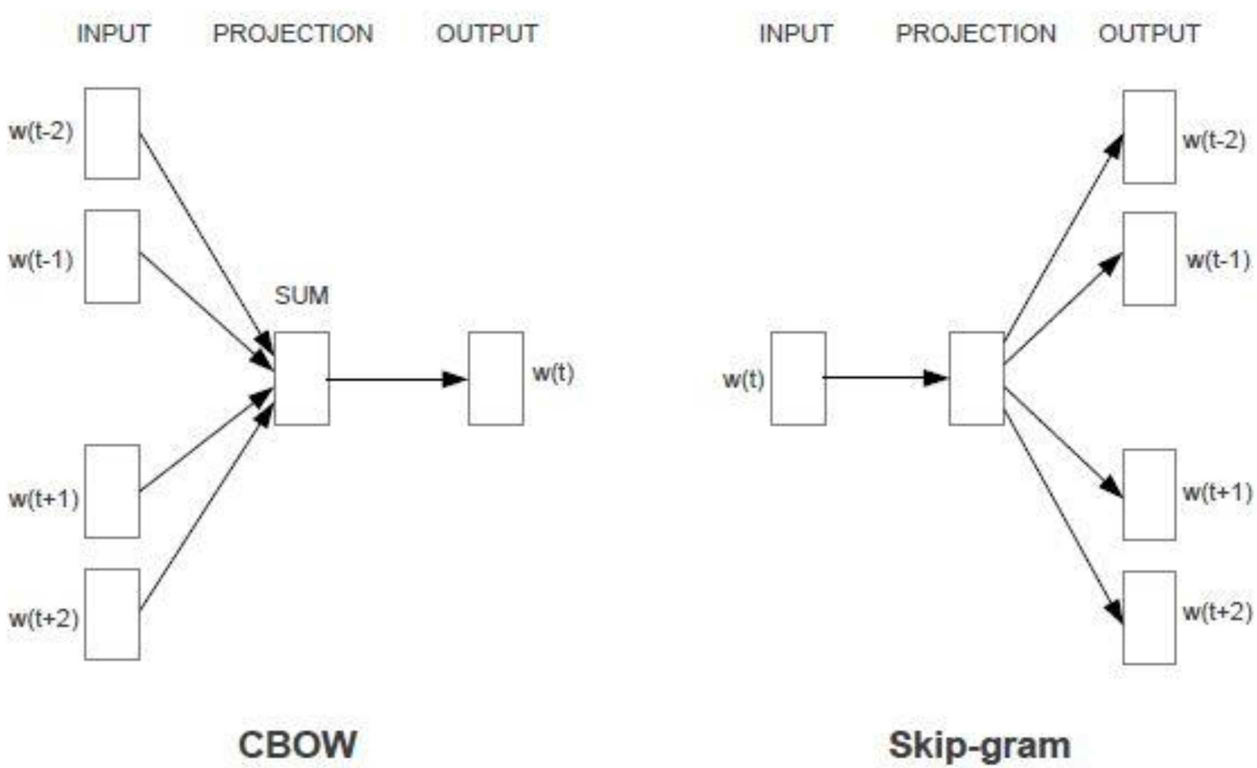
Microsoft Research

Learn RNN-LM by BackProp-Through-Time with gradient thresholding



During the training of RNN-LMs, the RNN unfolds into a deep feed-forward network (Ph.D. thesis of Mikolov, 2012).

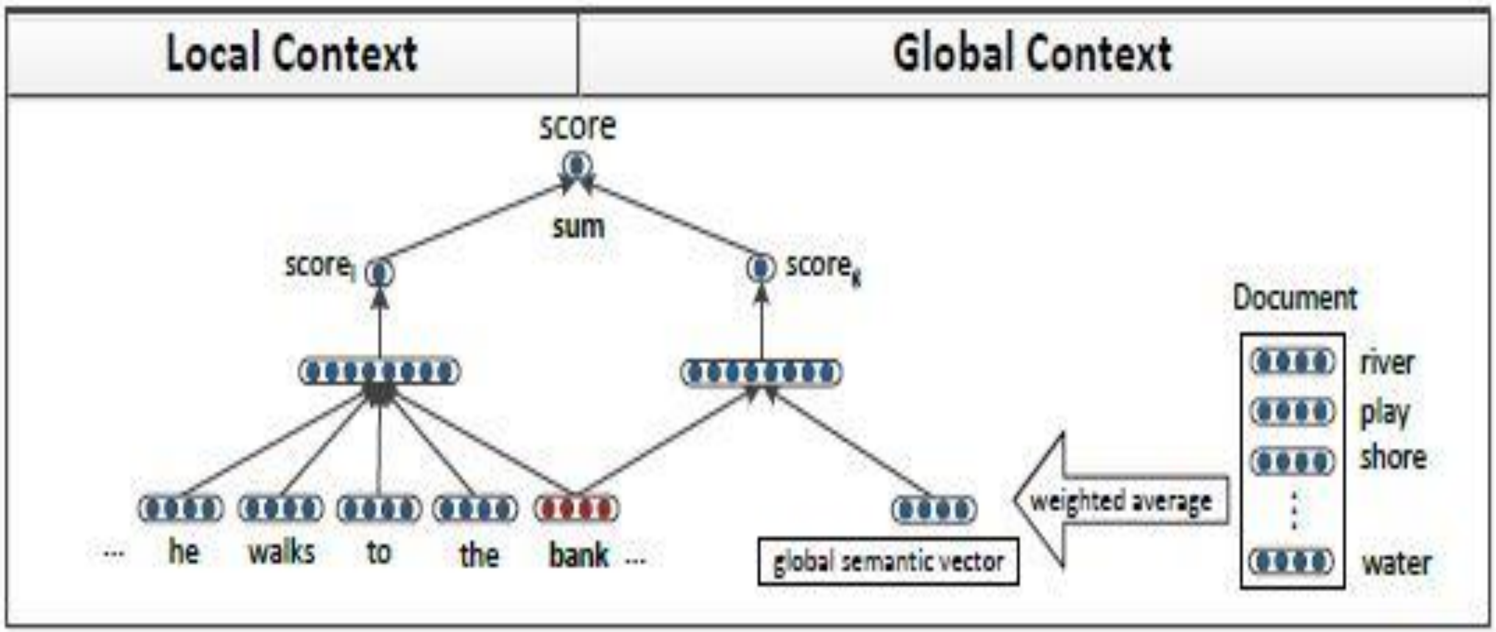
Deriving Word Embeddings



Continuous Bag-of-Words

The CBOW architecture (a) on the left, and the Skip-gram architecture (b) on the right. [after (Mikolov et al., 2013a), @ICLR].

Word-embedding model using **recursive neural nets** with local/global contexts



The extended word-embedding model using a recursive neural network that takes into account not only local context but also global context. The global context is extracted from the document and put in the form of a global semantic vector, as part of the input into the original word-embedding model with local context. Taken from Figure 1 of (Huang et al., 2012). *[after (Huang et al., 2012), @ACL]*.

Microsoft Research

Deep Visual Semantic Embedding Model

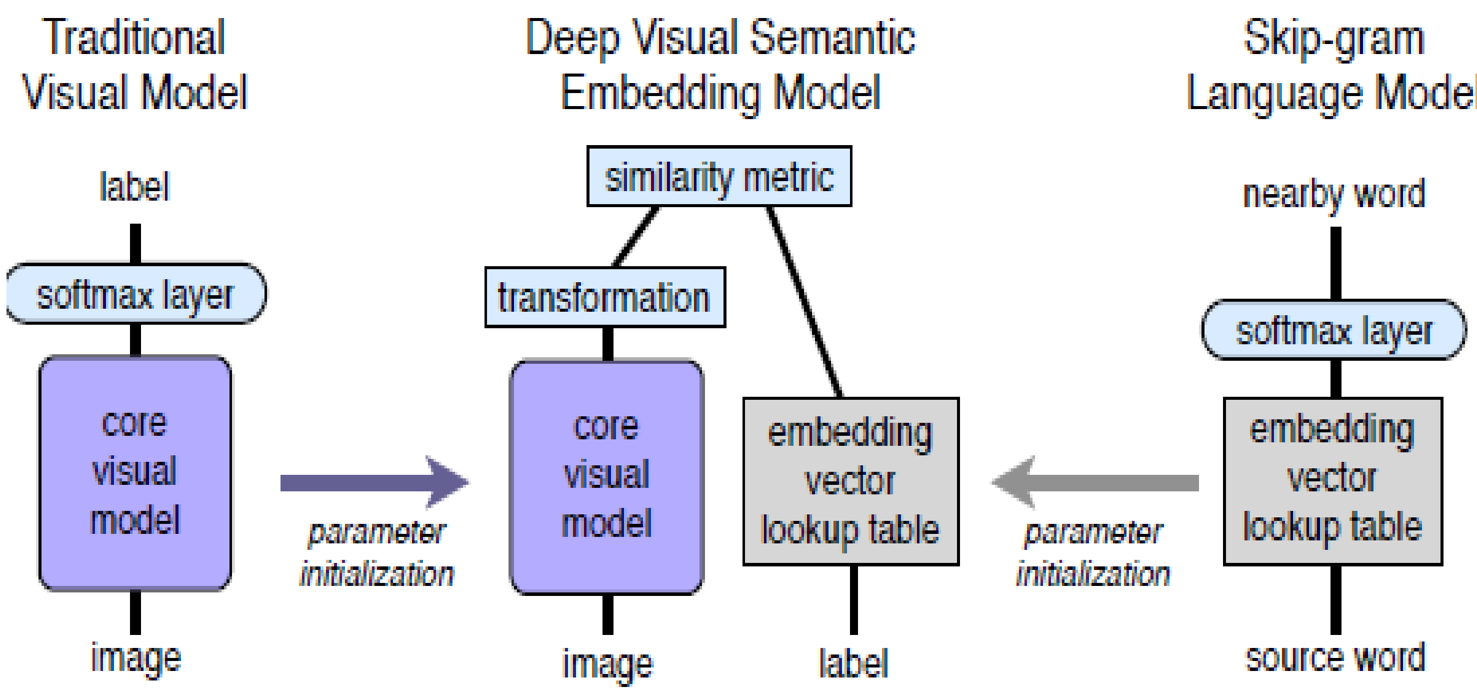
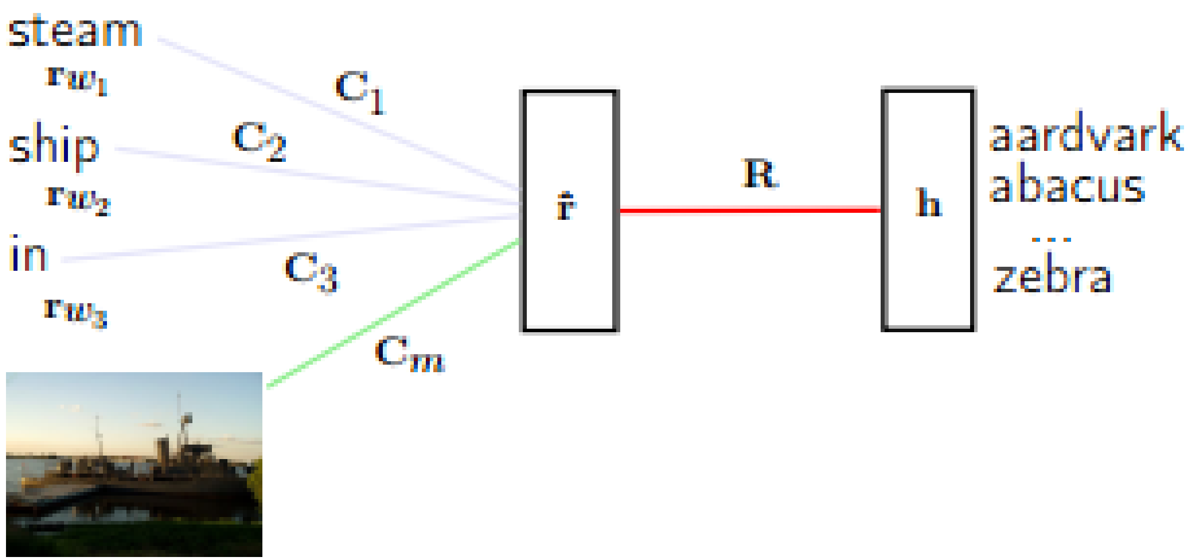


Illustration of the multi-modal DeViSE architecture. The left portion is an image recognition neural network with a softmax output layer. The right portion is a skip-gram text model providing word embedding vectors; see Chapter 8.2 and Figure 8.3 for details. The center is the joint deep image-text model of DeViSE, with the two Siamese branches initialized by the image and word embedding models below the softmax layers. The layer labeled "transformation" is responsible for mapping the outputs of the image (left) and text (right) branches into the same semantic space. [after (Frome, et al., 2013), @NIPS].

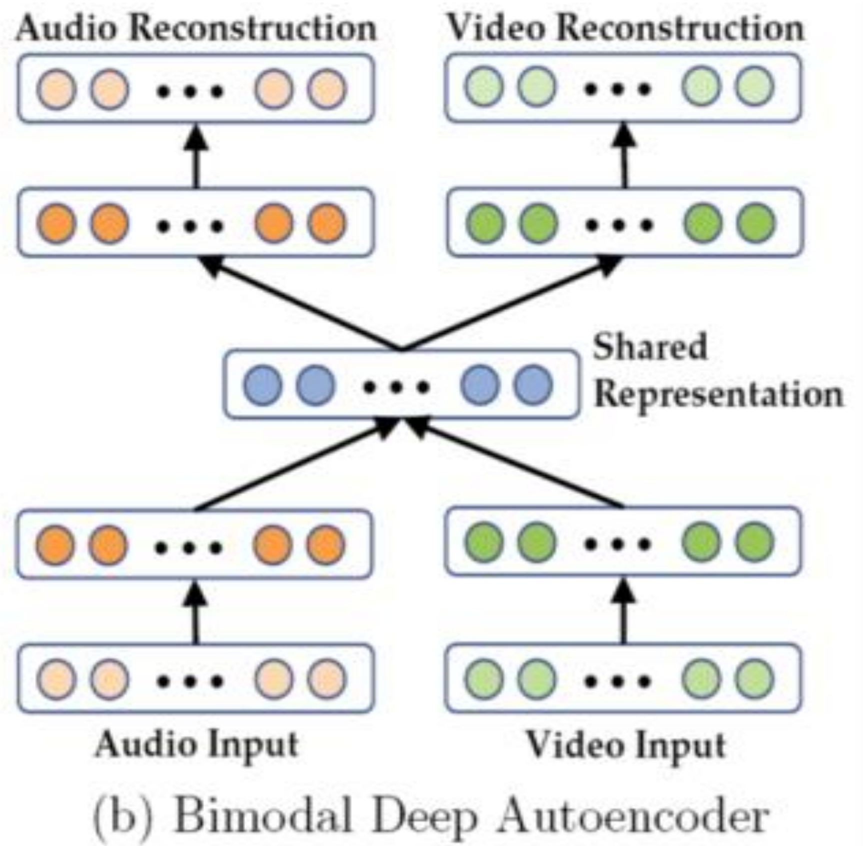
Multi-Modal Language Model



A multi-modal language model (of the type of log-bilinear) which predicts a word conditioned not only on the previous words in the sentence but also on images. The model operates on word embedding vectors. [after (Kiros et al., 2013), @NIPS].

Microsoft Research

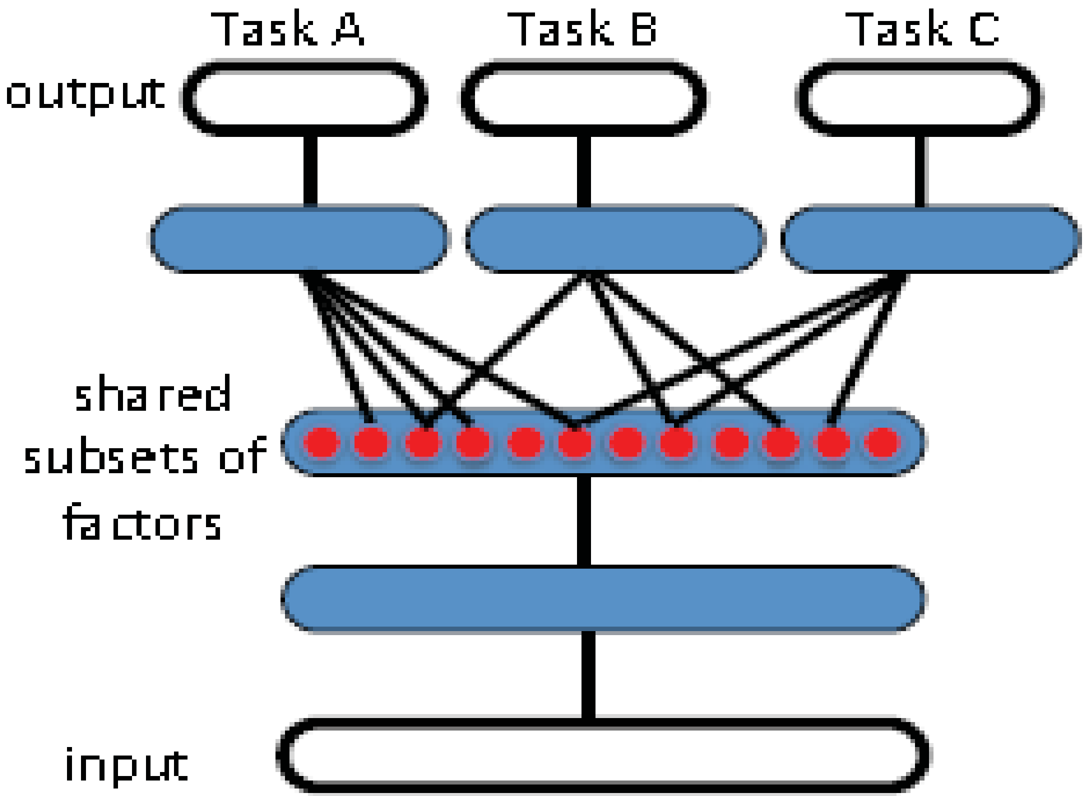
Multi-Modal Audio-Visual Deep Autoencoder



The architecture of a deep denoising autoencoder for multi-modal audio/speech and visual features. [after (Ngiam et al., 2011), @ICML].

Microsoft Research

A DNN for Multi-Task Learning

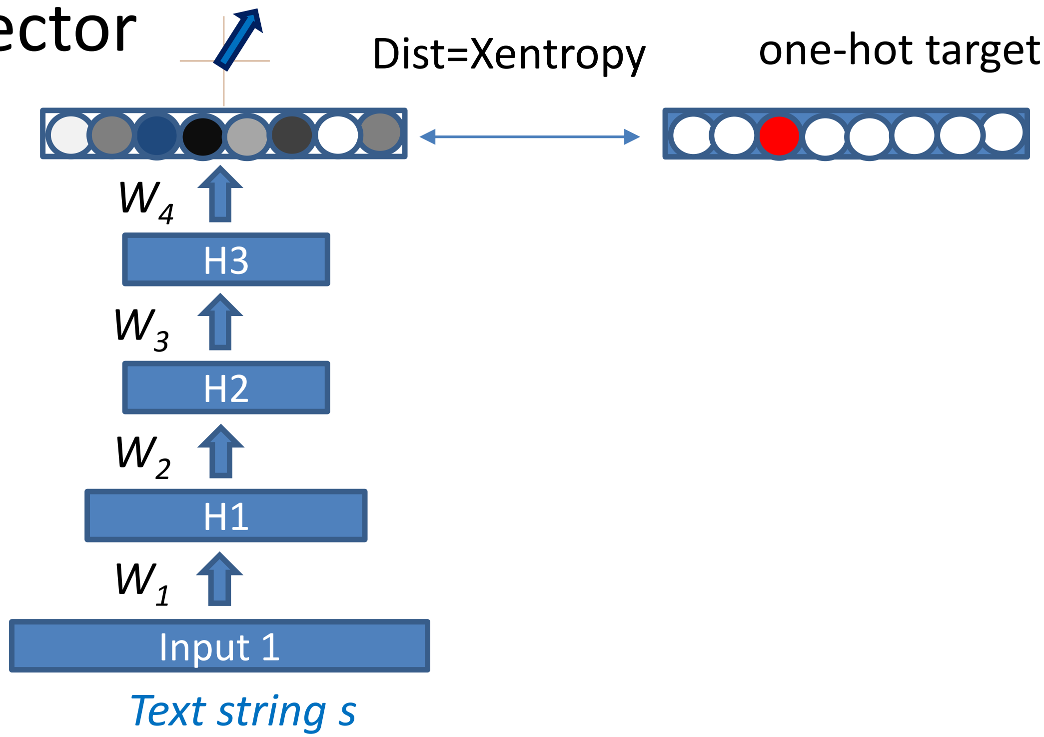


A DNN architecture for multitask learning that is aimed to discover hidden explanatory factors shared among three tasks A, B, and C. [after (Bengio, 2013), @IEEE].

Microsoft Research

From Common Deep Models to DSSM

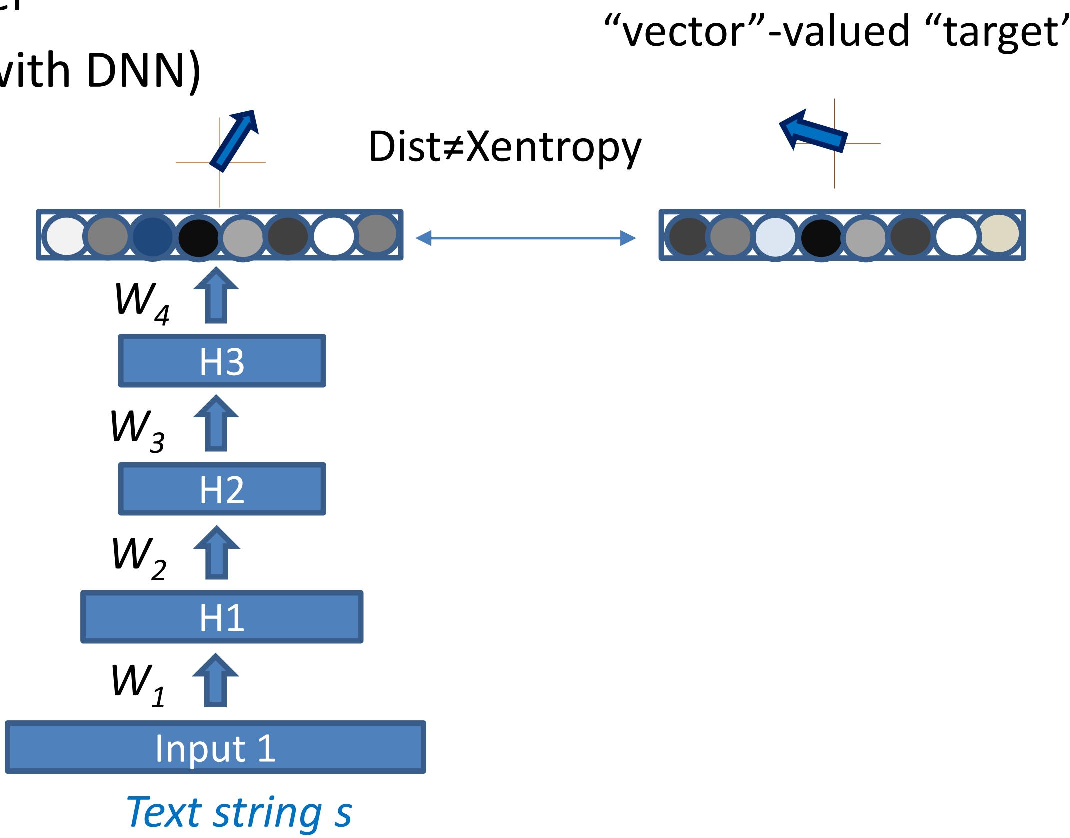
- Common deep models reviewed so far:
  - Mainly for classification
  - Target: one-hot vector
  - Example of DNN:



Microsoft Research

From DNN to DSSM

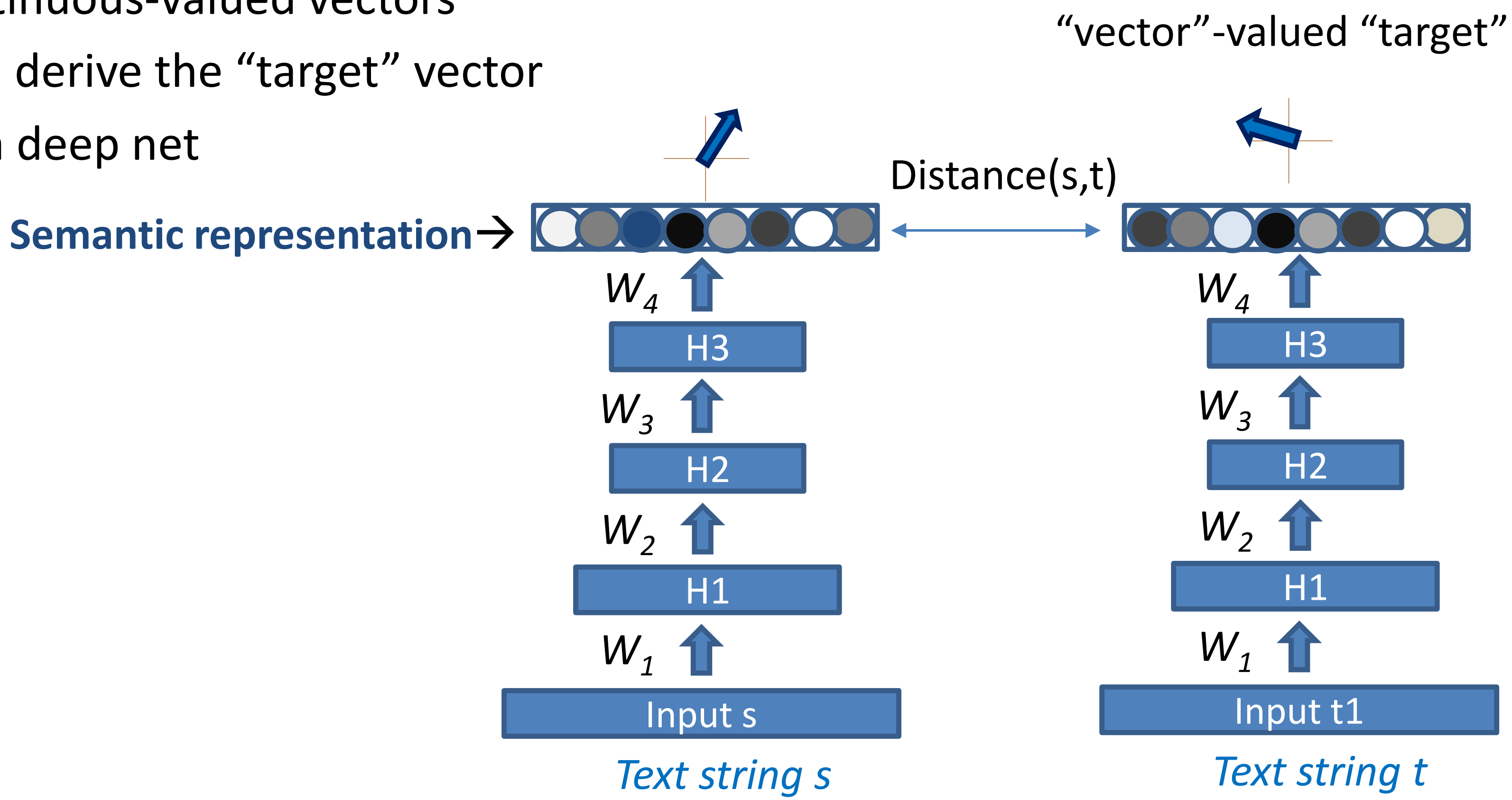
- DSSM
  - Deep-Structured Semantic Model, or
  - Deep Semantic Similarity Model
  - For ranking (not classification with DNN)
  - Step 1: target from "one-hot" to continuous-valued vectors



Microsoft Research

From DNN to DSSM

- To construct a DSSM
  - Step 1: target from "one-hot" to continuous-valued vectors
  - Step 2: derive the "target" vector using a deep net



Microsoft Research

From DNN to DSSM

- To construct a DSSM
  - Step 1: target from "one-hot" to a continuous-valued vector
  - Step 2: derive the "target" vector using a deep net
  - Step 3: normalize two "semantic" vectors & computer their similarity

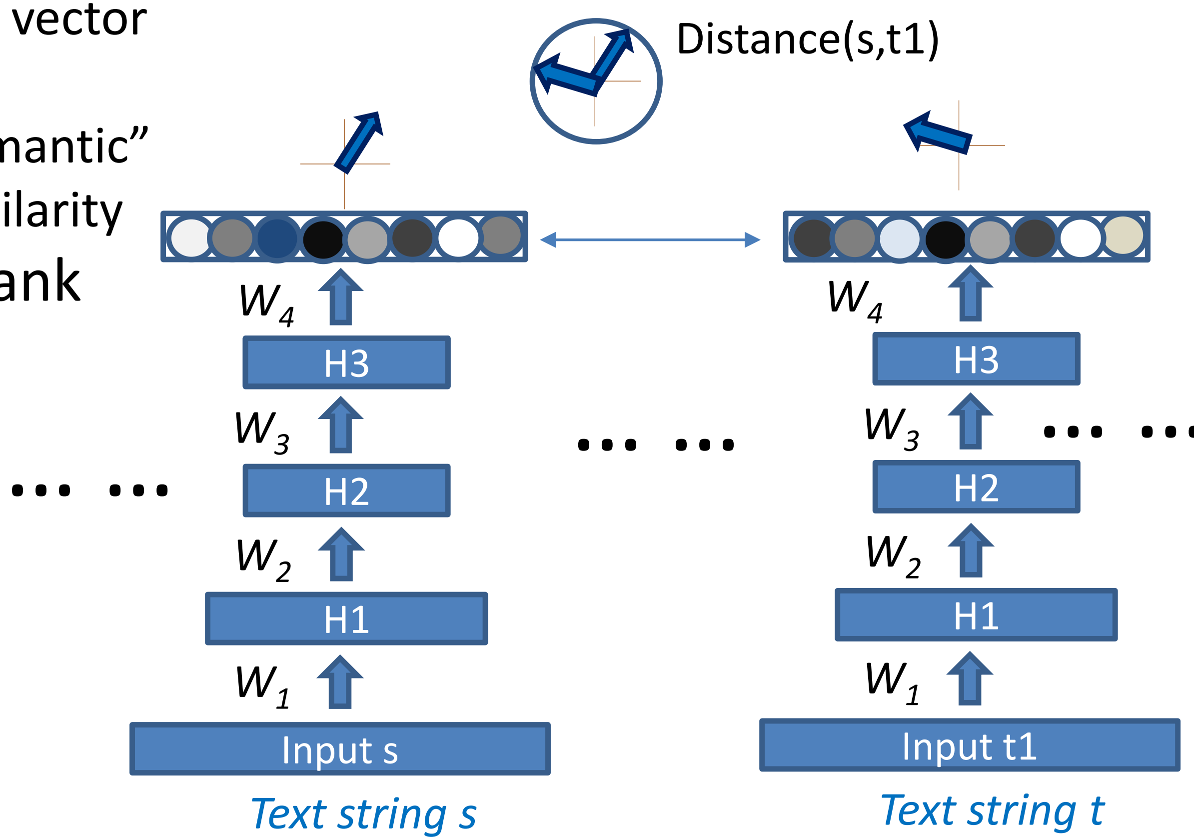
Use semantic similarity to rank documents/entities

cos(s,t1)

cos(s,t2)

cos(s,t3)

……



[Huang, He, Gao, Deng, Acero, Heck, "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data," in CIKM, Oct. 2013]

Microsoft Research

Interim Summary

- Common deep learning architectures
  - DNN (Deep Neural Nets), Tensor-DNN
  - CNN (Convolutional Neural Nets)
  - DSN (Deep Stacking Nets); Kernel-DSN, Tensor-DSN
  - RNNs (Recurrent and recursive Neural Nets)
- From DNN to DSSM (basic form)
- From DSSM to Conv-DSSM, Tensor-DSSM, Recurrent-DSSM, Kernel-DSSM, Stacking-DSSM
- The next 4 Parts will elaborate on the learning and applications of many of the above deep models

Part II

Deep learning in spoken language understanding

Background of SLU

- The three problems in spoken language understanding (SLU)
  - Domain classification
  - Intent detection
  - Semantic slot filling

"Show me flights from Boston to New York today"

Domain: travel

Intent: find_flight

"Show me flights from Boston to New York today"

Semantic slots: City-departure, City-arrival, Date

Why SLU is difficult?

- Huge variability in the spoken language
  - e.g., both the following two utterances are in the ***Travel*** domain, ***Find_Flight*** intent, and same semantic slots, but are uttered very differently

(1) "I want to fly from San Francisco to New York in a weekend"

(2) "Show me weekend flights from SFO to JFK"

Domain & Intent Classification

- A semantic utterance classification (SUC) problem
  - $\hat{C} = argmax_{\{C\}} P(C|X)$
  - Where
    - $C \in \{C_1, \dots, C_M\}$ belong to one of the M semantic categories (e.g., domain or intent)
    - X is the input utterance

Microsoft Research

SUC: Common methods

- Common raw features usually include
  - Word n-grams (n=1, 2, 3), e.g., bi-gram,

$$f^{BG}_{c,w_x w_y}(C_r, W_r) = \begin{cases} 1, & \text{if } c = C_r \wedge w_x w_y \in W_r \\ 0, & \text{otherwise.} \end{cases}$$

- Common classifiers
  - Logistic regression

$$P(C|W) = \frac{1}{Z} \sum_i w_i f_i(C, W)$$

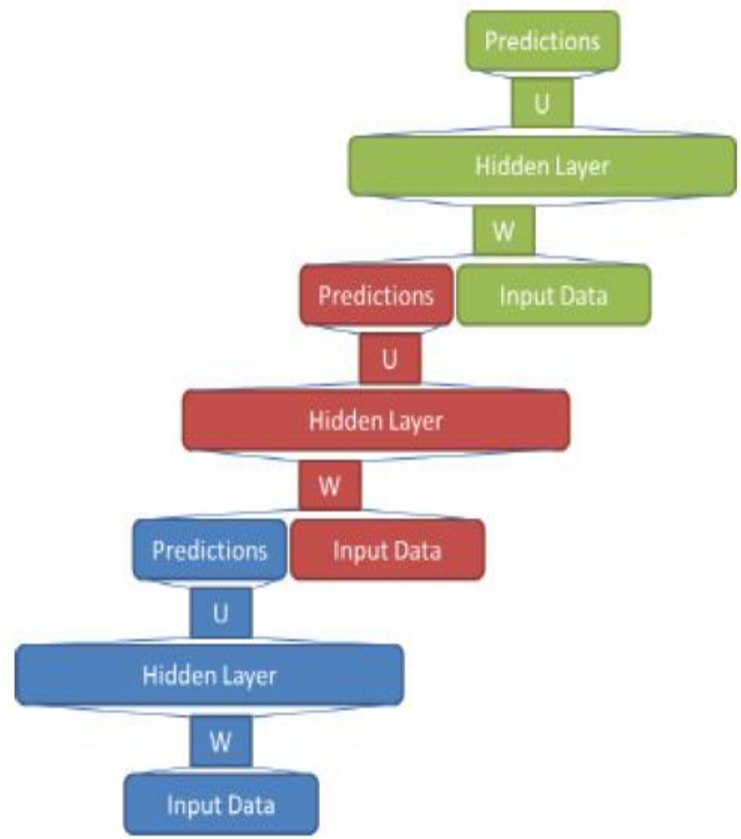
  - Boosting, SVM, etc.

Microsoft Research

SUC: Deep Convex Net

Deep convex net for semantic utterance classification:

1) A stack of a series of 3-layer perceptron modules
2) At each module
    1) Hidden layer is non-linear, other two are linear
    2) W is fixed (could be random valued or initialized by RBM)
    3) U is solved in closed-form – convex optimization
    4) No back-propagation
3) Output layer is concatenated with raw input to form input layer of the next module



[Tur, Deng, Hakkani-Tur, He, ICASSP2012]

Microsoft Research

SUC: Results

	No. Utt.	Avg. No. Words
Training	16,000	7.60
Development	2,000	7.66
Test	1,902	7.58

Table 1. Data sets used in the experiments.

Layer	Dev	Test
Chance (Majority)	77.45%	76.71%
Baseline (Boosting)	13.15%	13.35%
1	15.30%	15.29%
2	14.05%	13.14%
3	**13.45%**	**12.67%**
4	14.25%	13.77%
5	15.10%	14.45%

Table 2. Semantic classification error rates using deep convex nets with varying number of stacked DCN modules, compared to the Boosting baseline. RBM is used to initialize lowest-level network weights using the discriminative features selected by Boosting.

Model	Dev	Test
Baseline (Boosting)	10.70%	10.40%
DCN	11.50%	**10.09%**

Table 3. Semantic utterance classification error rates using optimal number of features for the Boosting baseline system.

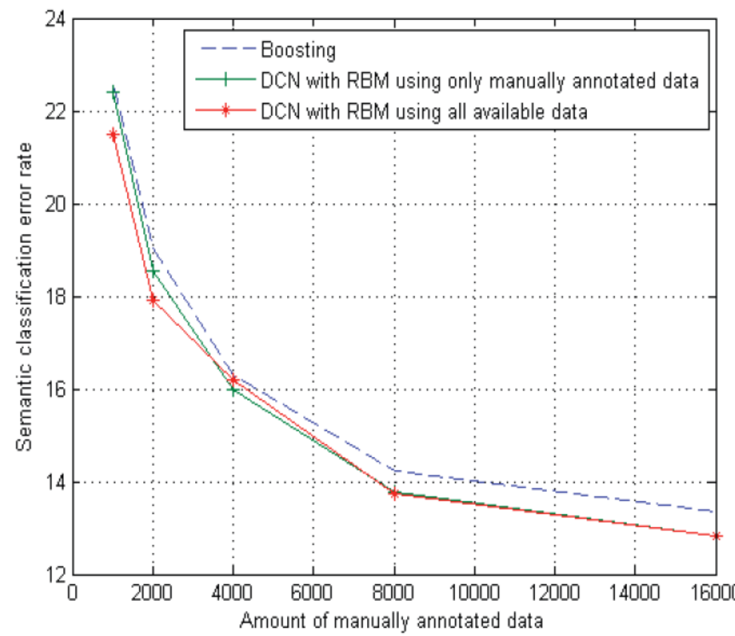


Fig. 2. Learning curves comparing Boosting with DCN with bottom layer initialized with RBM using only the annotated vs. all data.

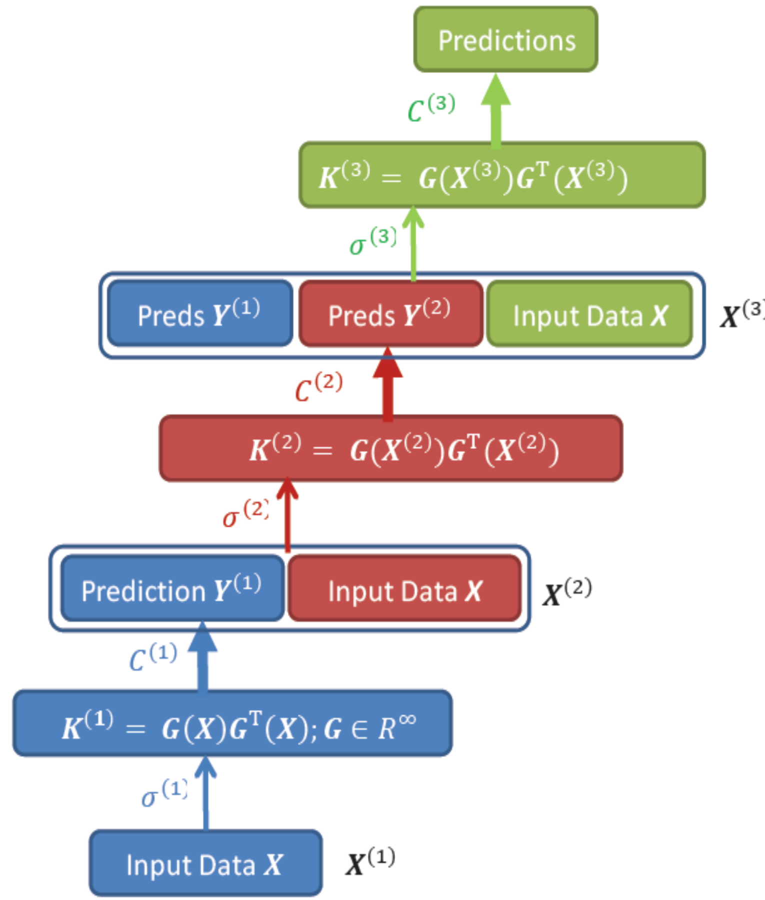
Microsoft Research

SUC: Kernel Deep Convex Net

Kernel Deep Convex Net :

1) Kernel version of DCN
2) No hidden layer, using kernel instead
   1) Gaussian kernel used
   2) Efficient learning – two hyper-parameters to train

(Deng, Tur, He, Hakkani-Tur, SLT2012)



Microsoft Research

Results

Table 2. *Comparisons of the domain classification error rates among the boosting-based baseline system, DCN system, and K-DCN system for a domain classification task. Three types of raw features (lexical, query clicks, and name entities) and four ways of their combinations are used for the evaluation as shown in four rows of the table.*

Feature Sets	Baseline	DCN	K-DCN
lexical features	10.40%	10.09%	**9.52%**
lexical features + Named Entities	9.40%	9.32%	**8.88%**
lexical features + Query clicks	8.50%	7.43%	**5.94%**
lexical features + Query clicks + Named Entities	10.10%	7.26%	**5.89%**

30% error reduction over a boosting-based baseline

Table 3. *More detailed results of K-DCN in Table 2 with Lexical+QueryClick features. Domain classification error rates (percent) on Train set, Dev set, and Test set as a function of the depth of the K-DCN.*

Depth	Train Err%	Dev Error%	Test Err%
1	9.54	12.90	12.20
2	6.36	10.50	9.99
3	4.12	9.25	8.25
4	1.39	7.00	7.20
5	0.28	6.50	5.94
6	0.26	**6.45**	**5.94**
7	0.26	6.55	6.26
8	0.27	6.60	6.20

Error keeps decreasing when up to six layers are added up

Semantic Slot Filling

A example in the Airline Travel Information System (ATIS) corpus

	show	*flights*	*from*	*boston*	*to*	*new*	*york*	*today*
Slots	O	O	O	B-dept	O	B-arr	I-arr	B-date

Slot filling can be viewed as a sequential tagging problem

Microsoft Research

Slot Filling: Common methods

Conditional random field (CRF)

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, \mathrm{x}_t^{(i)}) - \sum_{i=1}^{N}\log Z(\mathrm{x}^{(i)}) - \sum_{k=1}^{K}\frac{\lambda_k^2}{2\sigma^2}.$$

- N: number of training samples
- T: number of words in the sentence i
- K: “observation” functions (feature functions)
- x: input words in the sentence
- y: output tags

Other variants of CRF exist, e.g., semi-CRF.

Microsoft Research

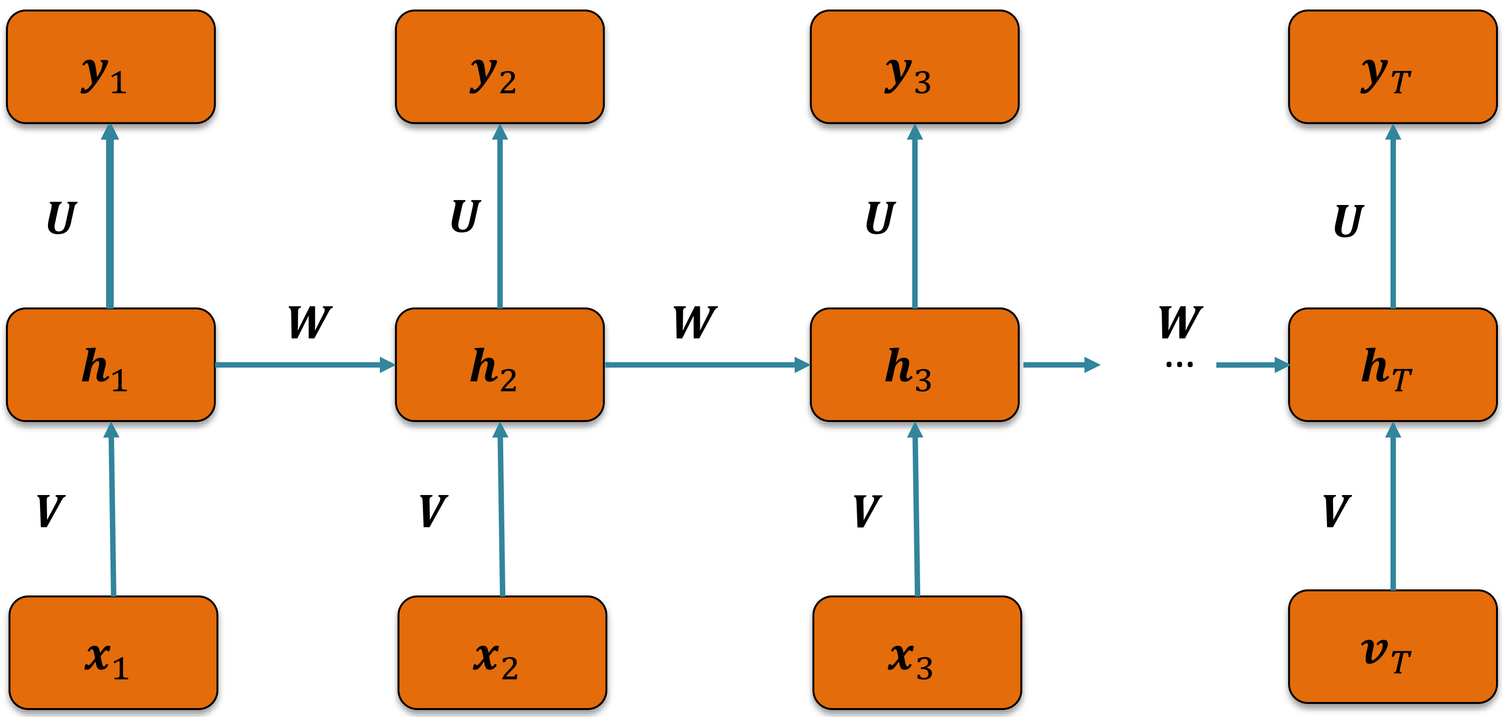
Recurrent Neural Networks for Slot Filling

- Using the (Elman-type) RNN for slot filling:

$y_t = SoftMax(U \cdot h_t), where\ h_t = \sigma(W \cdot h_{t-1} + V \cdot x_t)$

where x_t: $the\ input\ feature$, y_t: $the\ output\ tag$

h_t is the hidden layer that carries the information from time $0 \sim t$

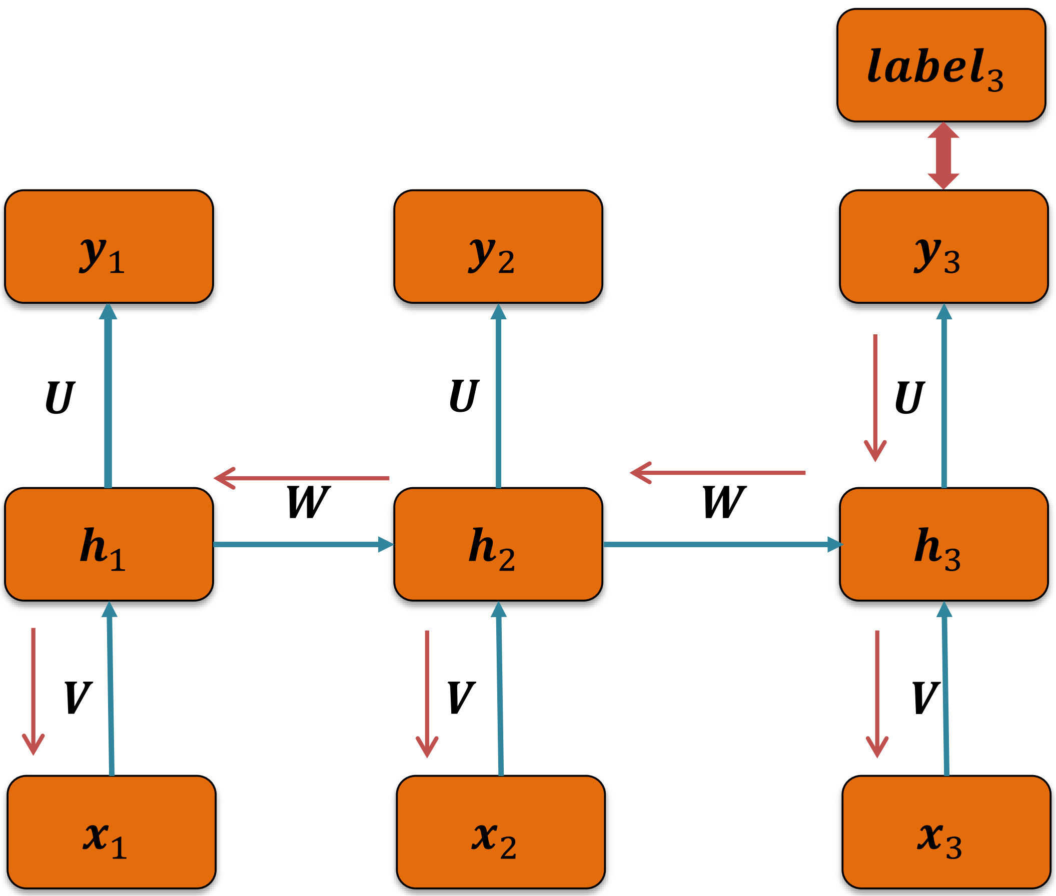


(Mesnil, He, Deng, Bengio, IS2013) (Yao, Zweig, Hwang, Shi, Yu, IS2013)

Microsoft Research

Training the RNN

- Back-propagation through time (BPTT):



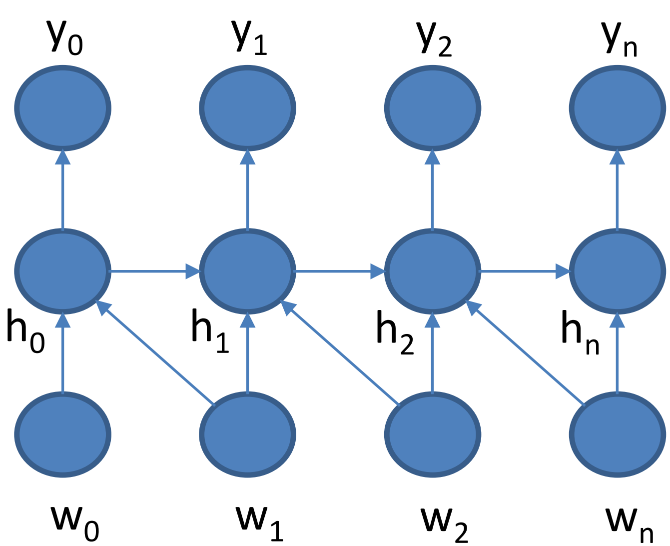
at time $t = 3$

1. Forward propagation
2. Generate output
3. Calculate error
4. Back propagation
5. Back prop. through time

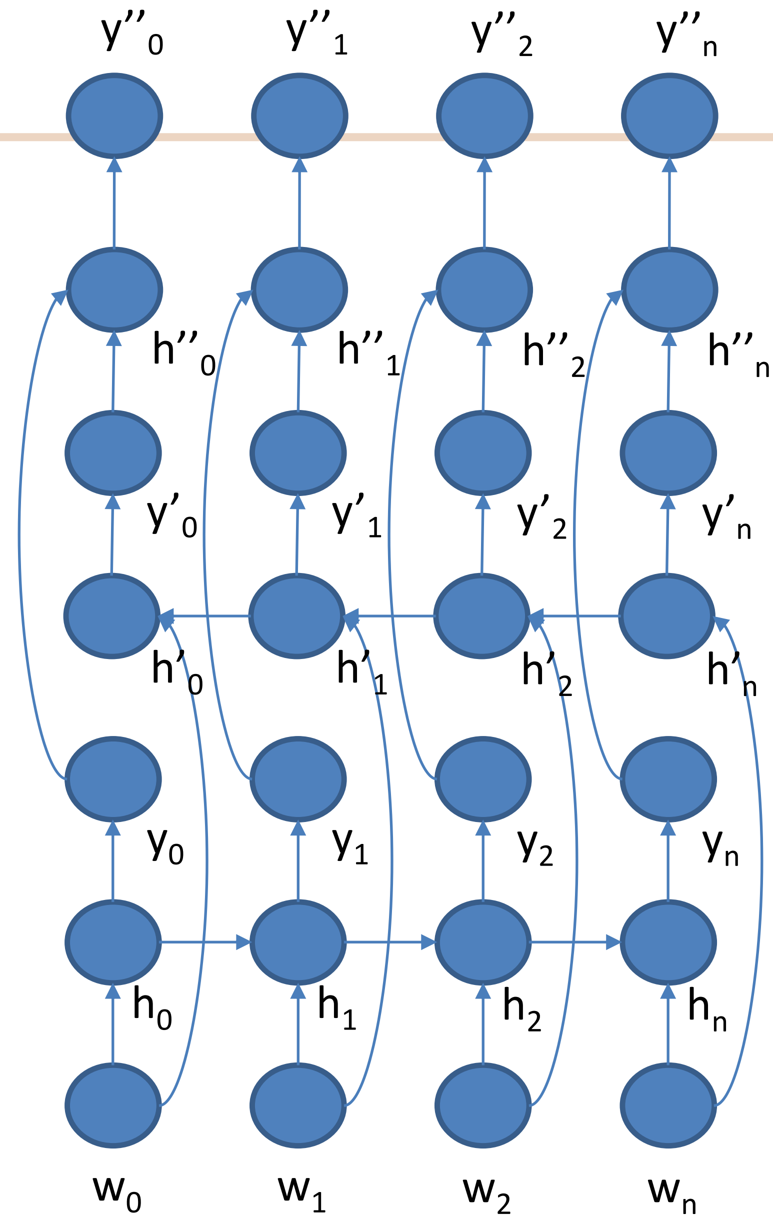
Variants of RNNs

Bi-directional Jordan RNN

RNN with look-ahead context



(Yao, Zweig, Hwang, Shi, Yu, IS2013)



(Mesnil, He, Deng, Bengio, IS2013)

Microsoft Research

Results

- Evaluated on the ATIS corpus
  - 4978 utterances for training
  - 893 utterances for testing
  - Using word feature only
  - Baseline CRF: 92.94% in F1-measure

~25% error reduction

SGD vs. minibatch training

With local context window

Model	Elman	Jordan	Hybrid
Stochastic GD	94.55 $\pm$0.51	94.66 $\pm$0.23	94.75 $\pm$0.31
Sentence-minibatch	94.54 $\pm$0.23	94.33 $\pm$0.19	94.25 $\pm$0.28

Left-to-right vs. bi-directional RNN

With local context window

Model	Elman	Jordan
Left-to-right	94.54	94.33
bi-direction	94.73	94.03

Without local context window

Model	Elman	Jordan
Left-to-right	93.15	65.23
bi-direction	93.46	90.31

Interim Summary

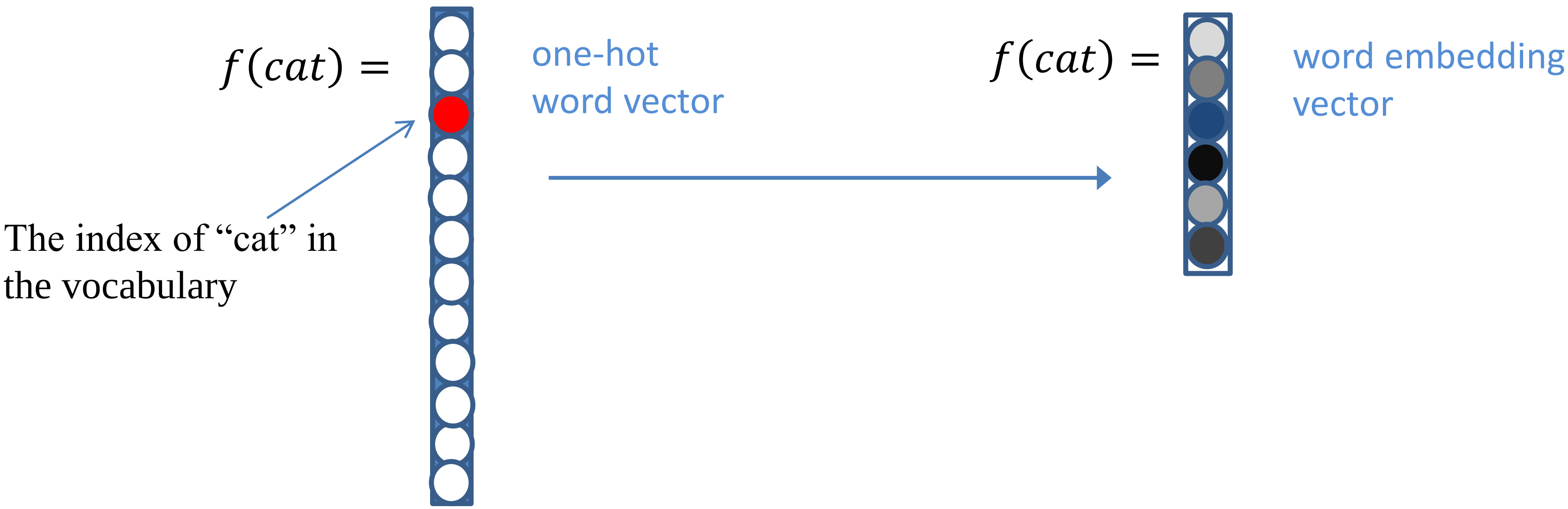
- Introduction to SLU
- DNN/DCN/K-DCN for Domain/intent detection
- RNN and its variants for slot filling
- Deep learning models demonstrate superior performances on these tasks

Part III

Learning Semantic Embedding

Word Embedding

- Word embedding
  - A low-dimensional continuous vector representation for each word
  - Captures the word meaning in a semantic space



- Common neural network based word embedding approaches
  - SENNA embedding
  - NN/RNN language model based embedding
  - CBOW & Skip-gram

SENNA embedding

Scoring:

$$Score(w_1, w_2, w_3, w_4, w_5) = U^T \sigma(W[f_1, f_2, f_3, f_4, f_5] + b)$$

Training:

$$J = \max(0, 1 - S^+ + S^-)$$ Update the model until $S^+ > 1 + S^-$

Where

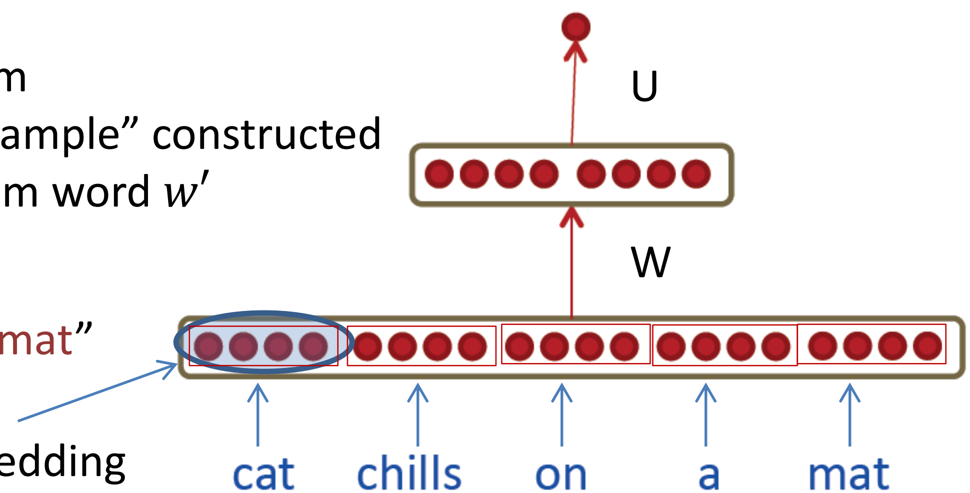
$$S^+ = Score(w_1, w_2, w_3, w_4, w_5)$$
$$S^- = Score(w_1, w_2, w', w_4, w_5)$$

And

$< w_1, w_2, w_3, w_4, w_5 >$ is a valid 5-gram

$< w_1, w_2, w', w_4, w_5 >$ is a "negative sample" constructed by replacing the word w_3 with a random word w'

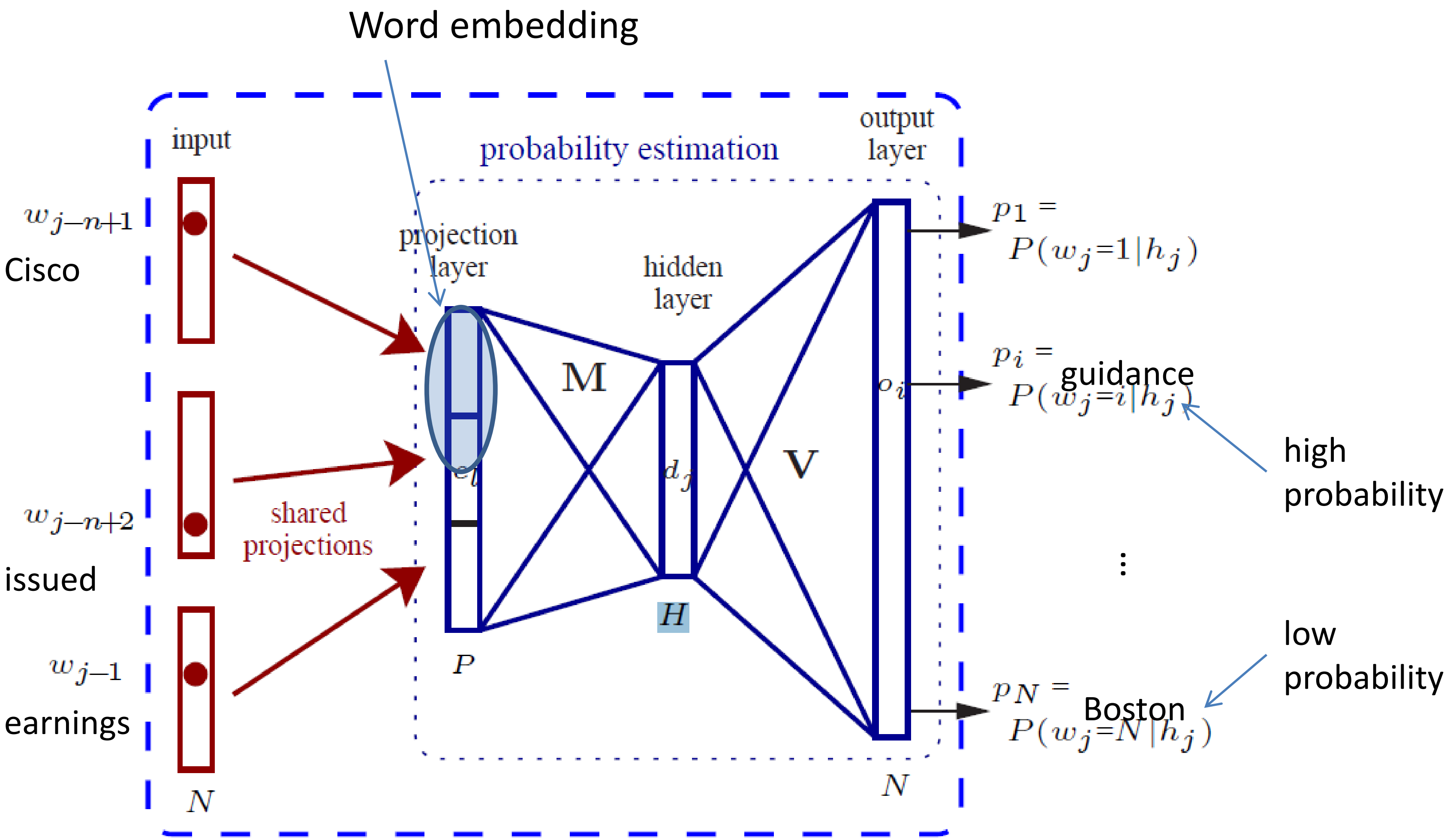
e.g., a negative example: "cat chills X a mat"



Word embedding

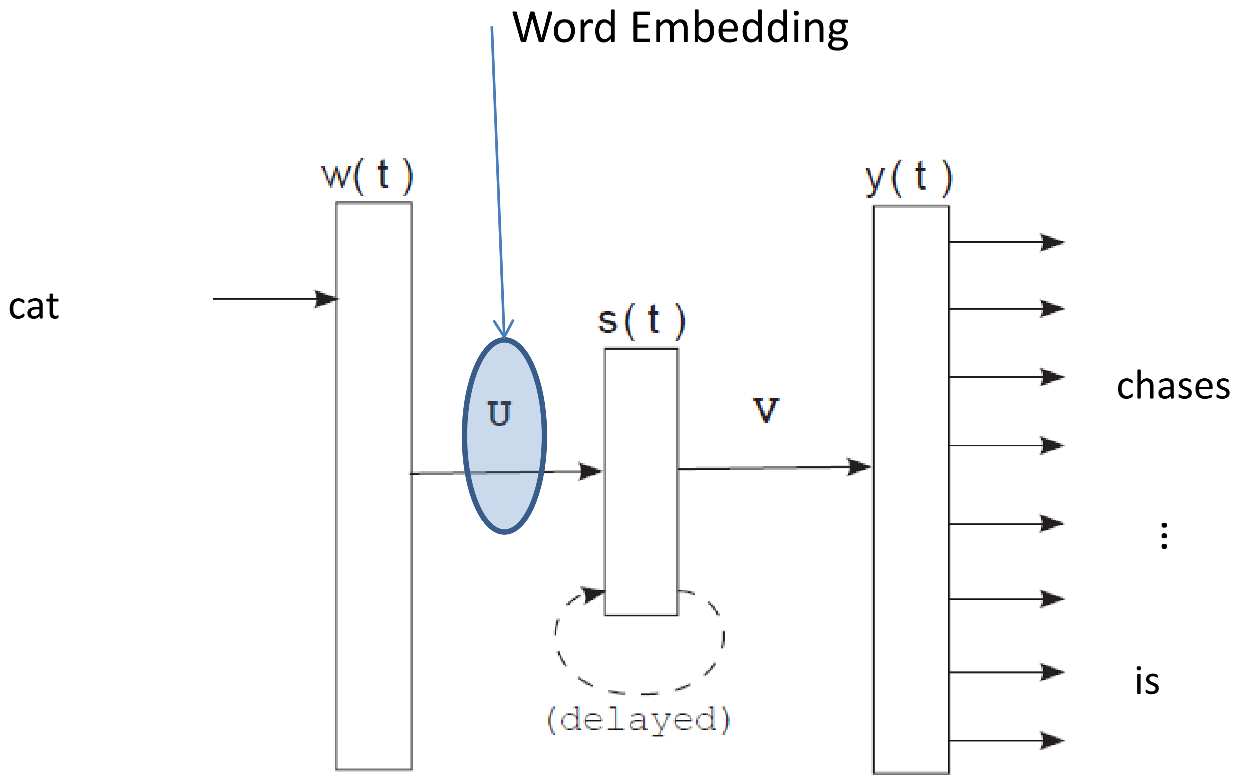
(Collobert et al., JMLR 2011)

NN-LM based word embedding



From Schwenk et al., 2006

RNN-LM base word embedding

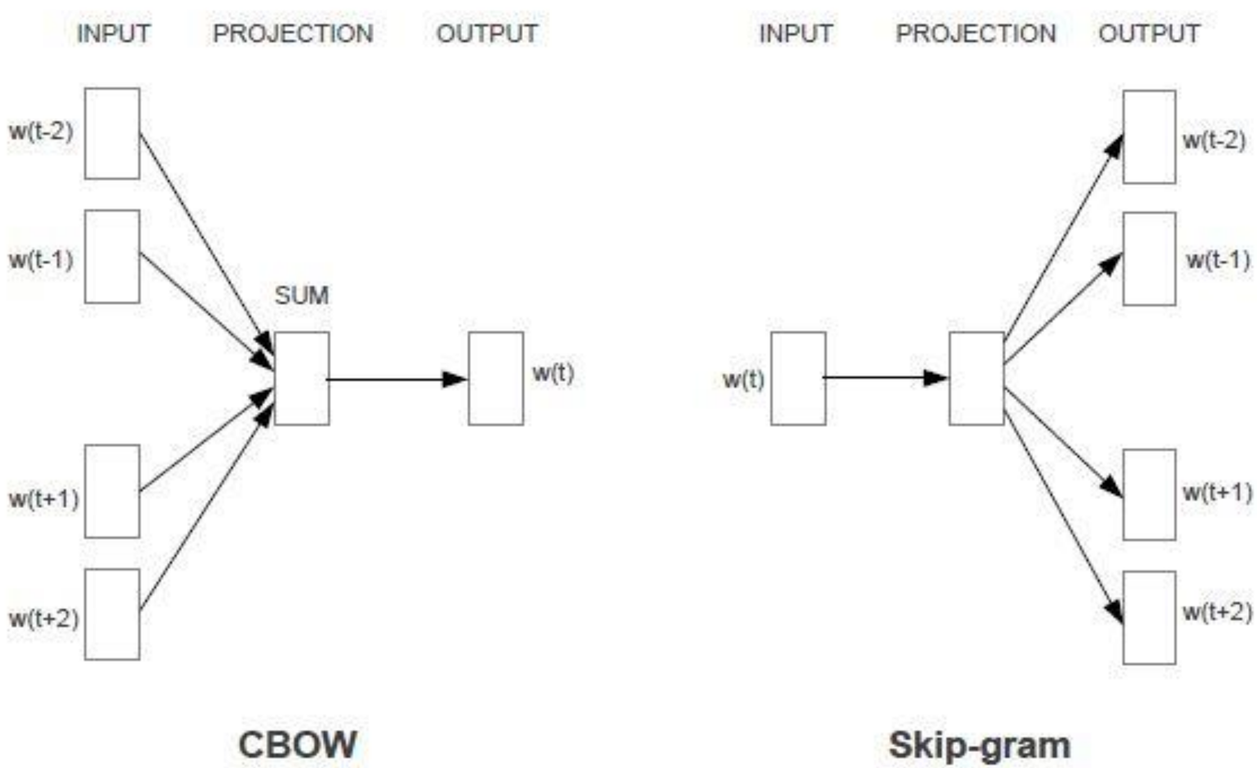


RNN::FFNN <----> IIR-Filter::FIR-Filter

Mikolov et al., 2011

Microsoft Research

CBOW/Skip-gram Word Embeddings



Continuous Bag-of-Words

The CBOW architecture (a) on the left, and the Skip-gram architecture (b) on the right. [after (Mikolov et al., 2013a), @ICLR].

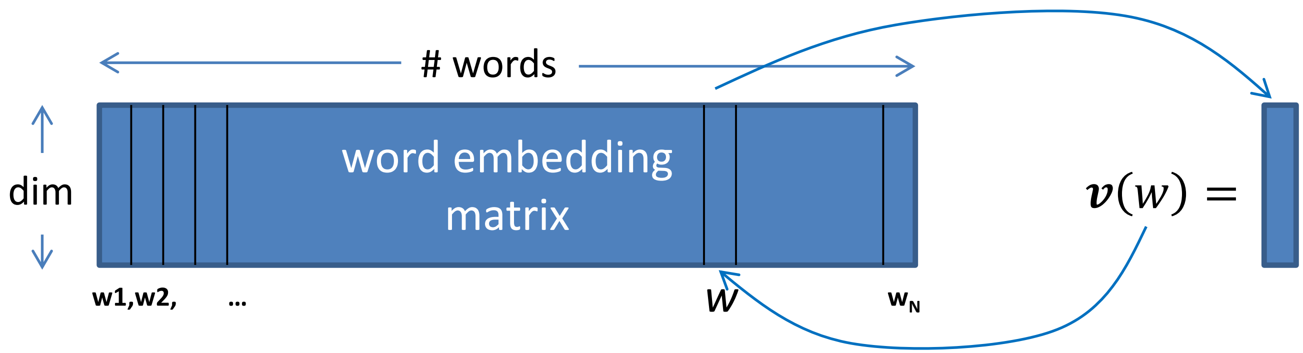
Training of Word Embedding

- These word embedding models are trained in an unsupervised, but discriminative, way
  - They are trained solely on text data
  - They are trained trying to make the score of a valid word n-gram higher than that of negative samples
    - Raw features come from the context of the word
    - SENNA tries to make the prediction score of the "true" 5-gram higher than others with a random word in the middle
    - NN/RNN LMs try to make the prediction score of the "true" next word higher than other words
    - CBOW tries to make the prediction score of the "true" central word higher than others

Microsoft Research

Word Embedding: Revisit

- Word embedding is a neat and effective representation:



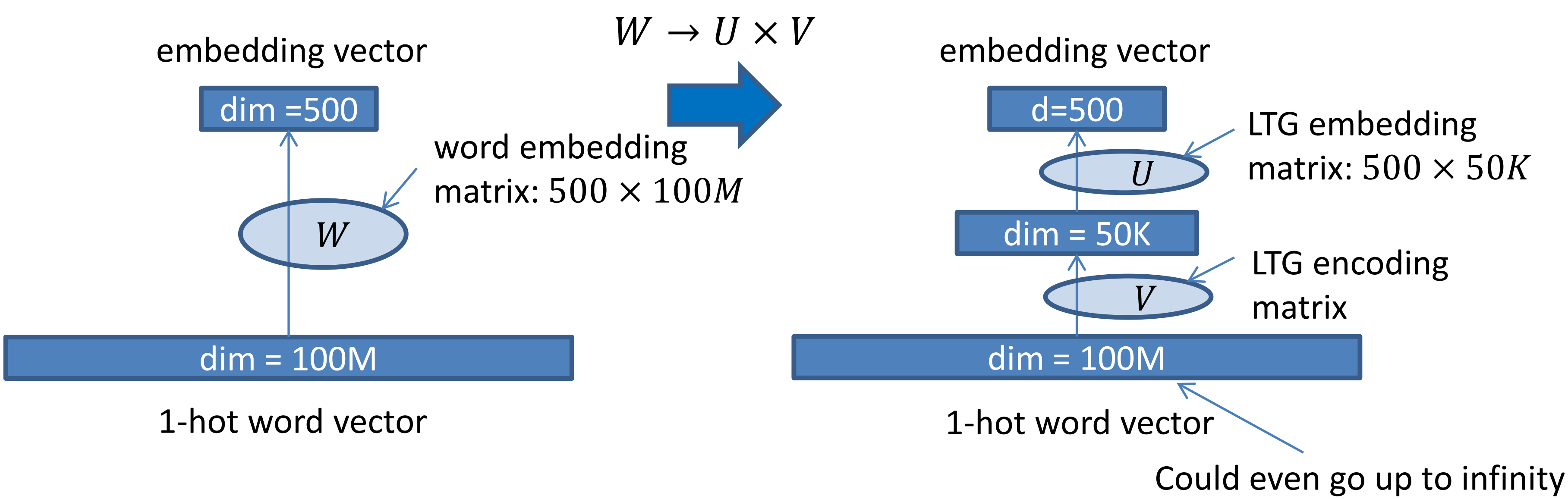
- A decomposable, robust representation is preferable for large scale NL tasks
  - Vocabulary of real-world big data tasks could be huge (*scalability*)

    >100M unique words in a modern commercial search engine log, and keeps growing
  - New words, misspellings, and word fragments frequently occur (*generalizability*)

Microsoft Research

From Word Embedding to Sub-word Embedding

- Learning sub-word embedding
  - Learn embedding on sub-word units, such as letter-trigram (LTG)
    - E.g., cat → #cat# → #-c-a, c-a-t, a-t-#
  - Reduce the problem of modeling from an almost unbounded variability (word) to a bounded variability (sub-word)
    - E.g., there are only ~50K letter-trigrams (37^3)



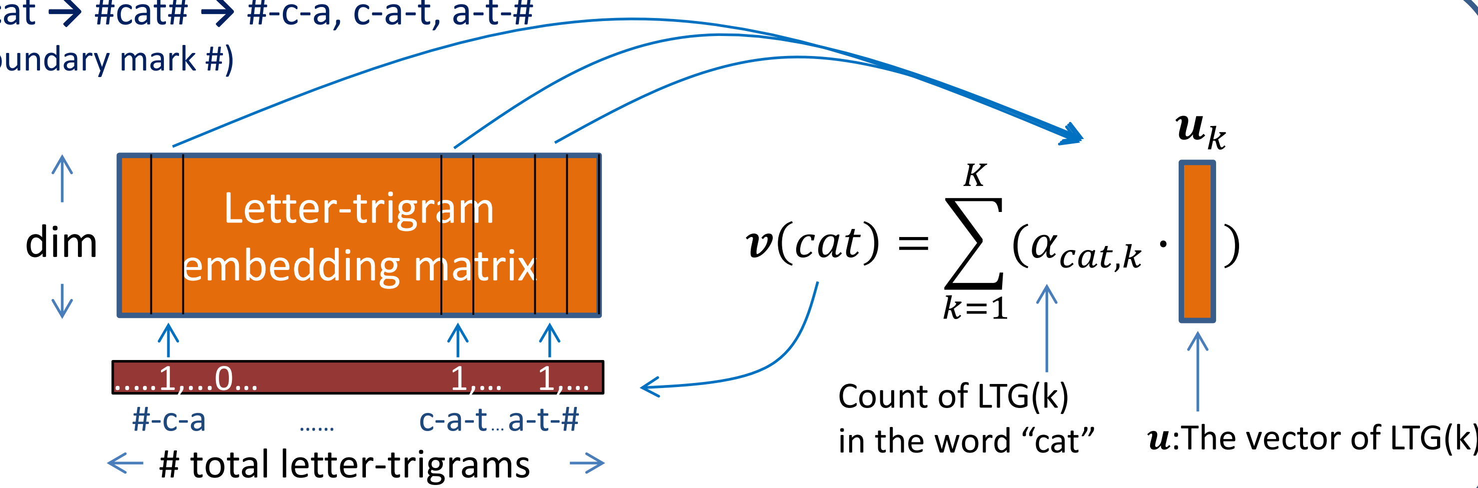
[Huang, He, Gao, Deng, Acero, Heck, 2013]

Microsoft Research

Letter-trigram as the Sub-word Unit

- Learn **one vector per letter-trigram** (LTG), the encoding matrix is a fixed matrix
  - Use the count of each LTG in the word for encoding

Example: cat → #cat# → #-c-a, c-a-t, a-t-#
(w/ word boundary mark #)



- Address both the *scalability* and *generalizability* issues

[Huang, He, Gao, Deng, Acero, Heck, 2013]

Letter-trigram based word representation

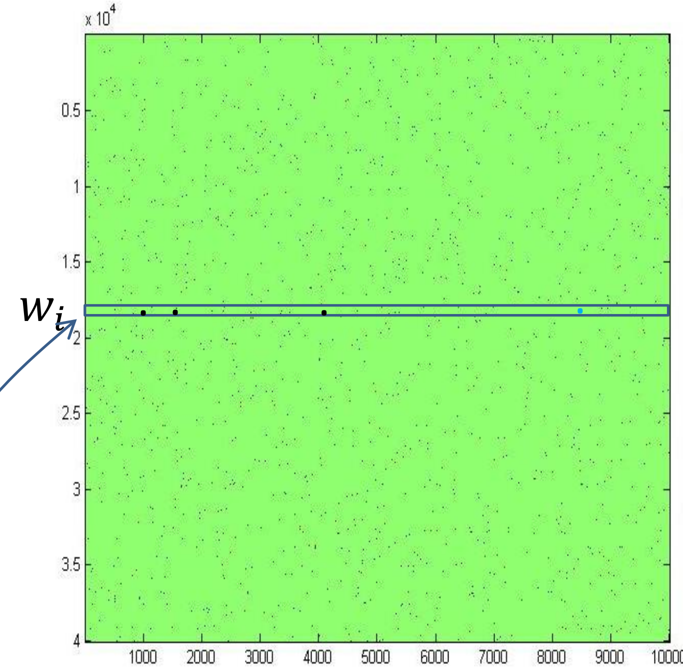
- Collision:
  - Different words have the same letter-trigram representation?
  - Statistics
    - collision rate ≈ 0.004%
    - Collision Example: #bananna# <=> #bannana#

Vocabulary size	Unique letter-trigram observed	Number of Collisions
40K	10,306	2
500K	30,621	22
5M	49,292	179

Other representation: random projection

- Sparse random projection matrix R with entries sampled i.i.d. from a distribution over [0, 1, -1]
- Entries of 1 and -1 are equally probable
- $P(R_{ij} = 0) = 1 - \frac{1}{\sqrt{d}}$, where d is the original input dimensionality.

[Li, Hastie, and Church 2006]



Each word will have a set of sparse random encoding of the 10000 basic units

Microsoft Research

More Word Input Representations

- Multi-hashing approach to input representation
- letters, context-dept letters, phones, context-dept phones, roots/morphs, context-dept morphs
- Word-level hashing

Semantic embedding: from word to phrase/doc

- A semantic representation at the phrase, sentence, or even document level is desirable
  - The meaning of a single word is often ambiguous.
  - A phrase/sentence/document contains rich contextual information that could be leveraged.
  - The semantic intent is better defined at the phrase/sentence level rather than at the word level.

Semantic embedding for phrases and documents

- History: Latent Semantic Analysis (LSA)
    - LSA extracts low dimensional semantic structure using SVD to get a low rank approximation of the word-document co-occurrence matrix
- Many extensions exist: PLSA, LDA, etc.
- However, the expressive power of linear models are restricted

- Go deeper:
    – e.g., semantic hashing (Salakhutdinov & Hinton 2007, 2010)

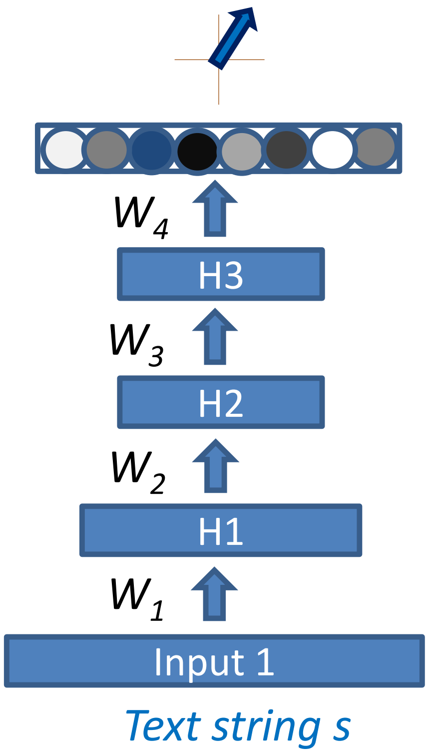
Microsoft Research

Deep models for phrase semantic embedding

Abstract representation in the semantic space

each layer gradually extracts deeper invariance

Raw text feature, e.g., bag-of-words.

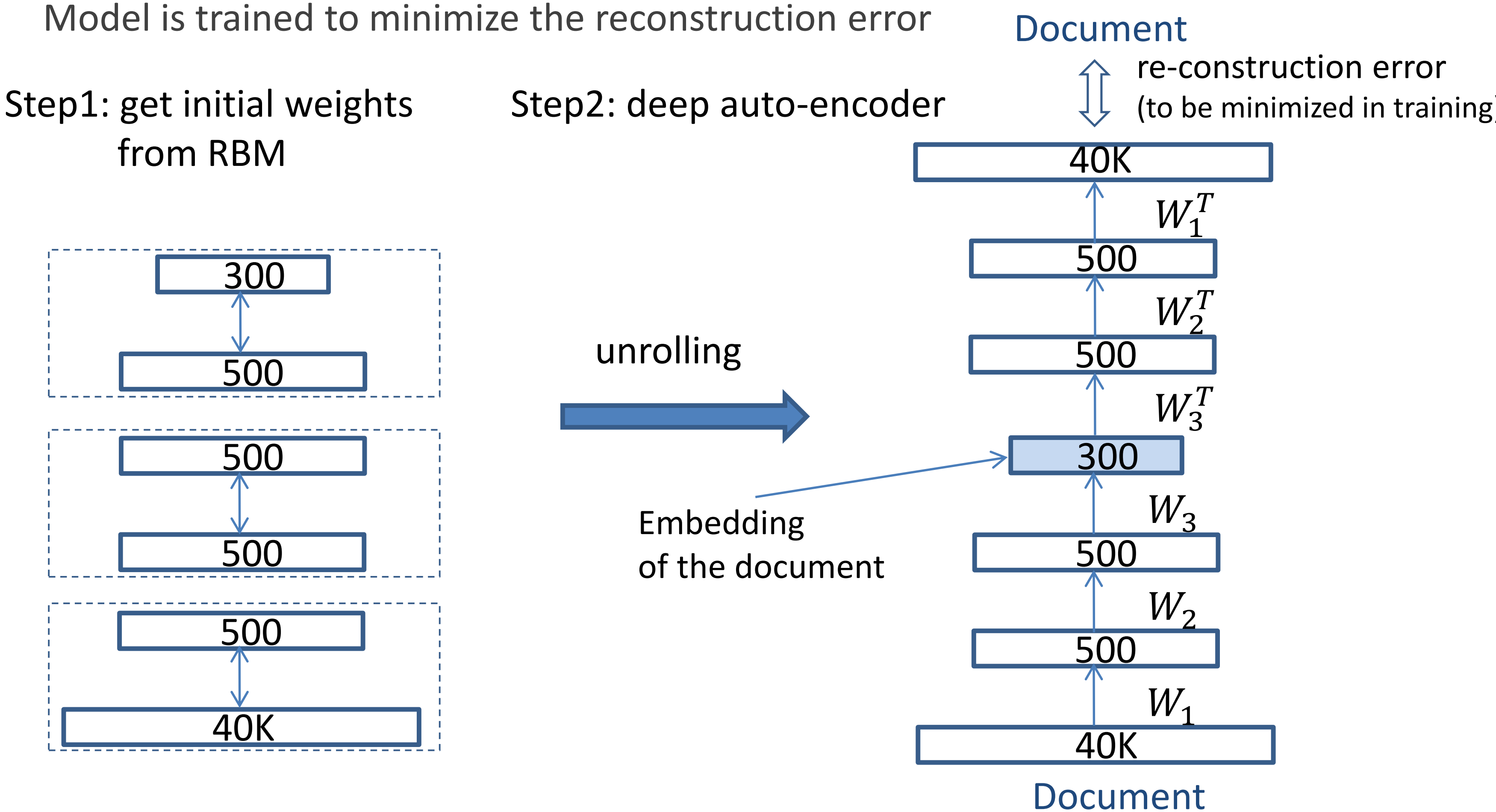


Semantic Hashing

1) Single layer learning: Restricted Boltzmann Machine (RBM)

2) Multi-layer training: deep auto-encoder, learn internal representations

Model is trained to minimize the reconstruction error



(Salakhutdinov & Hinton 2007, 2010)

Microsoft Research

Issues of the auto-encoder

- The objective for training the auto-encoder?
  - What is the relation between minimizing re-construction error and good embedding?
- What does *good embedding* mean?
  - Good embedding helps end-to-end tasks, so:
    - Optimizing embedding directly instead of minimizing the doc re-construction error
    - Learning the model with end-to-end user behavior log data (weak supervision) beside documents

Learning Semantic Embedding using the DSSM

- Deep structured semantic models (DSSM)
  - The DSSM refers to a series of **deep** semantic models developed recently at MSR
    - With variations on model structures and training objectives
  - The DSSM is trained by an **embedding similarity-driven objective**
    - projecting semantically similar phrases to vectors close to each other
    - projecting semantically different phrases to vectors far apart
  - The DSSM uses the **letter-trigram** sub-word embedding for the input word representation

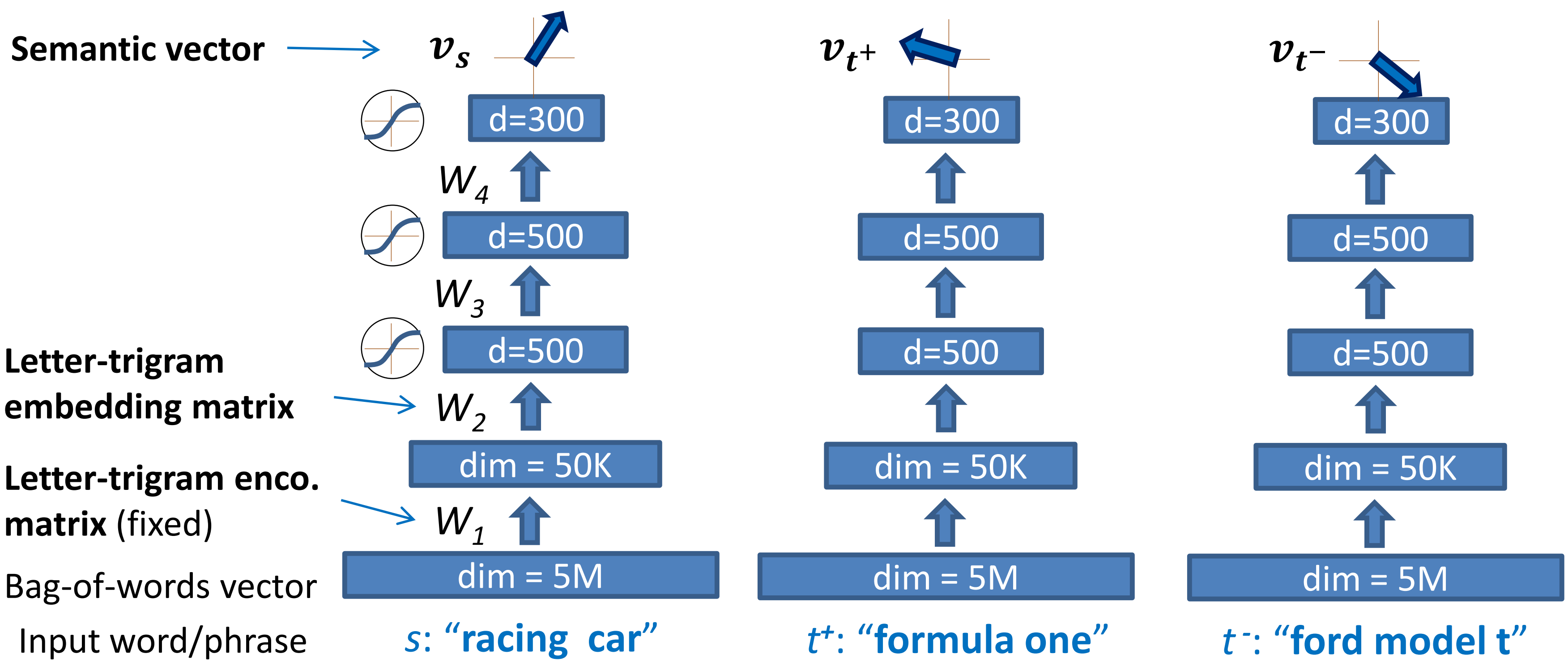
[Huang, He, Gao, Deng, Acero, Heck, 2013]
[Shen, He, Gao, Deng, Mesnil, 2014]

Learning Semantic Embedding using the DSSM

[Huang, He, Gao, Deng, Acero, Heck, 2013]

Initialization:

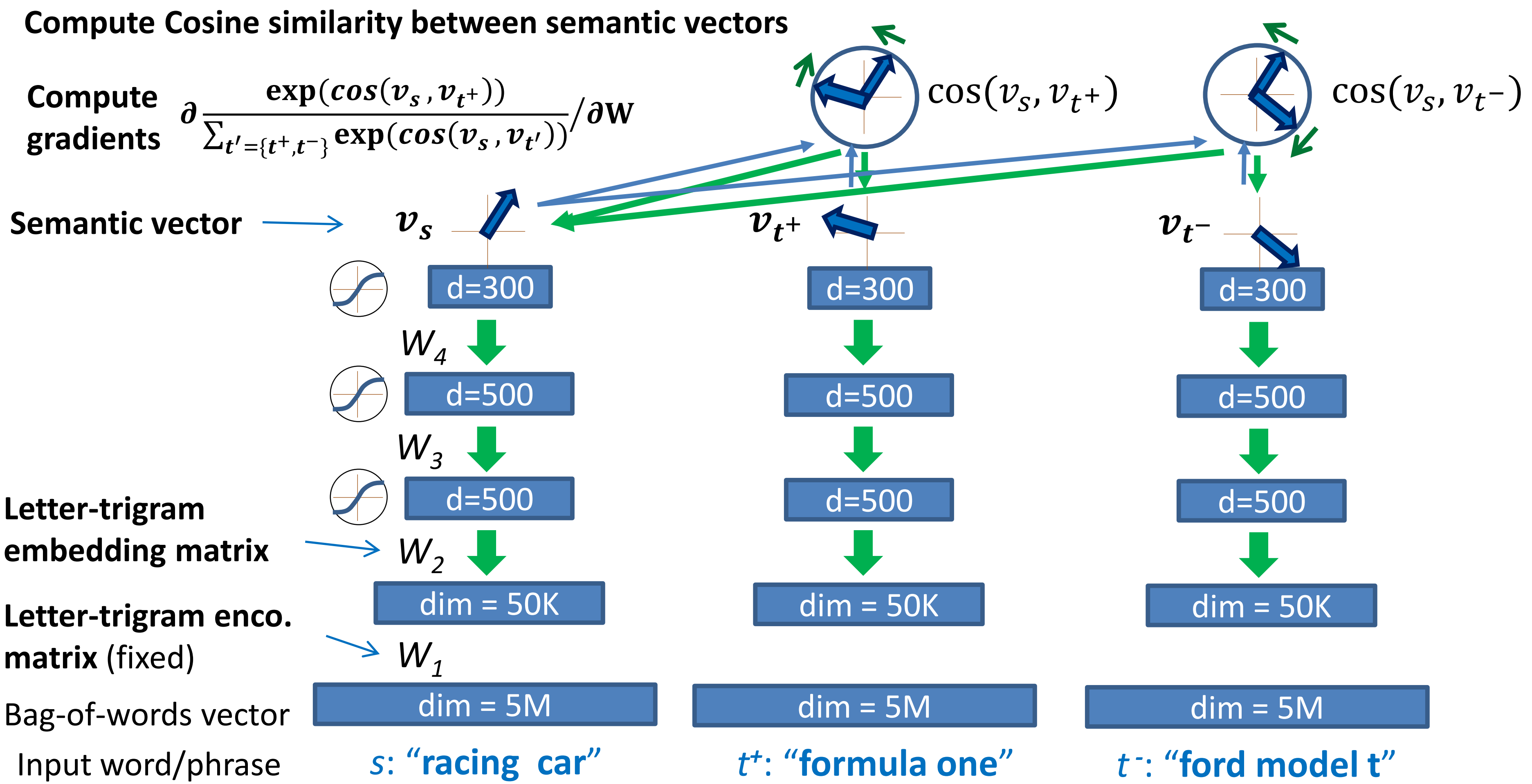
Neural networks are initialized with random weights



Learning Semantic Embedding using the DSSM

Training (Back Propagation):

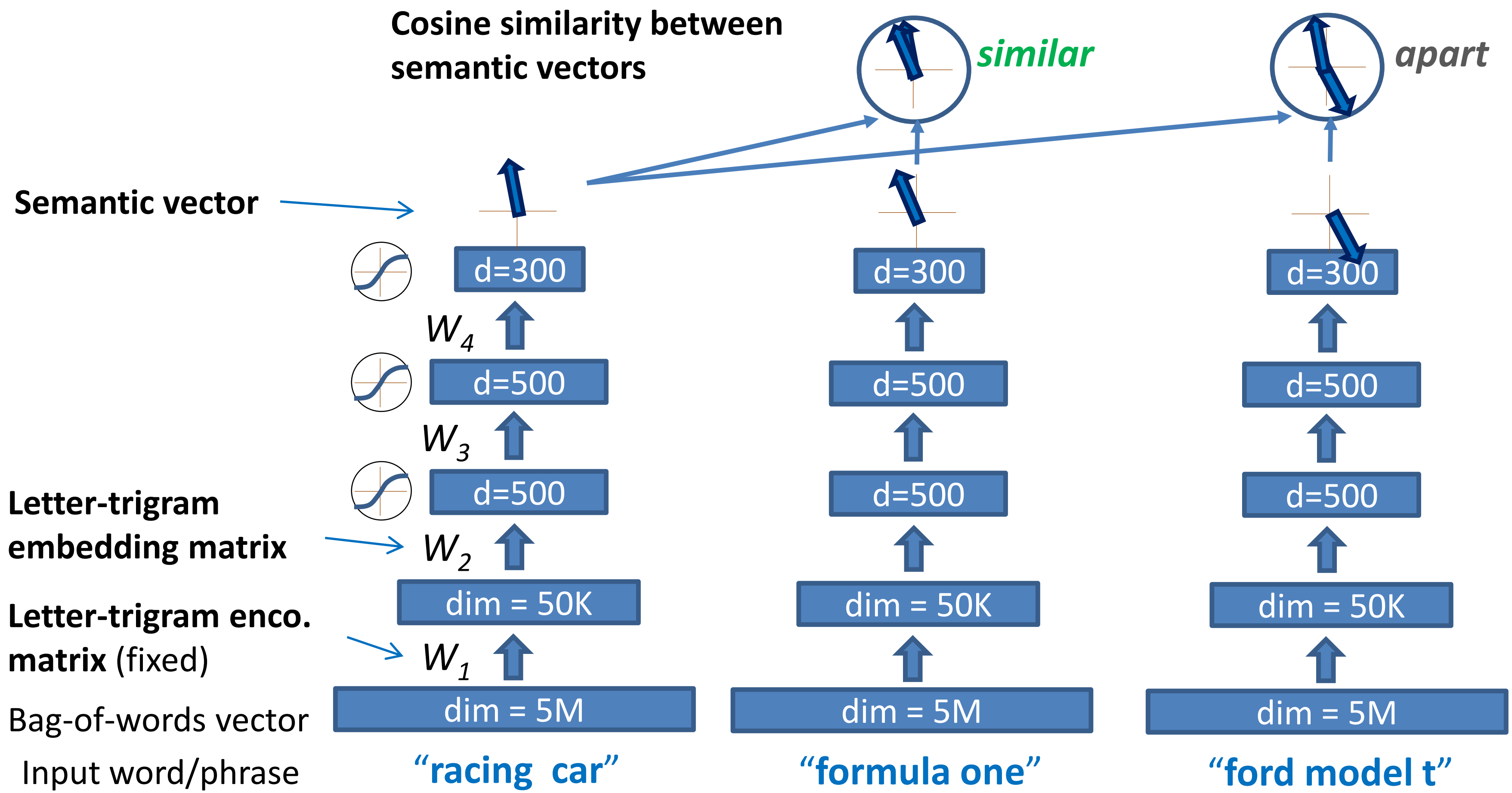
[Huang, He, Gao, Deng, Acero, Heck, 2013]



Learning Semantic Embedding using the DSSM

[Huang, He, Gao, Deng, Acero, Heck, 2013]

After training converged:



Evaluation

- Evaluated on a document ranking task
  - Docs are ranked by the cosine similarity between embedding vectors of the query and the doc

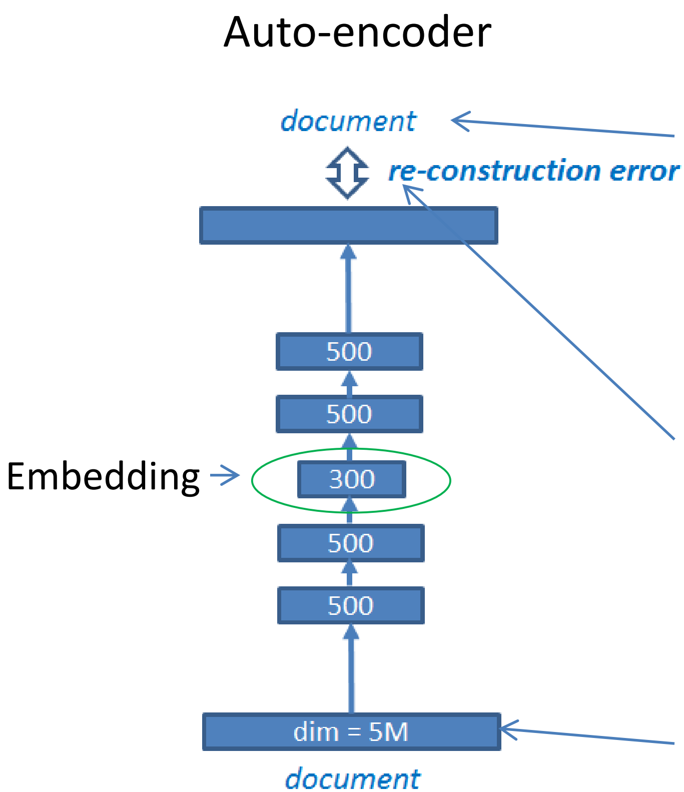
Model	Input dimension	NDCG@1 %
BM25 baseline	--	30.8
Probabilistic LSA (PLSA)		29.5
Auto-Encoder (Word)	40K	31.0 (+0.2)
DSSM (Word)	40K	34.2 (+3.4)
DSSM (Random projection)	30K	35.1 (+4.3)
DSSM (Letter-trigram)	30K	36.2 (+5.4)

DSSM-based embedding improves 5~7 pt NDCG over shallow models

The higher the NDCG score the better, 1% NDCG difference is statistically significant.

- The DSSM learns superior semantic embedding
- Letter-trigram + the DSSM gives superior results

Analysis of Auto-encoder vs. DSSM



Supervision:

AE: unsupervised
(e.g., doc<->doc)

DSSM: weakly supervised
(e.g., query<->doc search log)

Training objective:

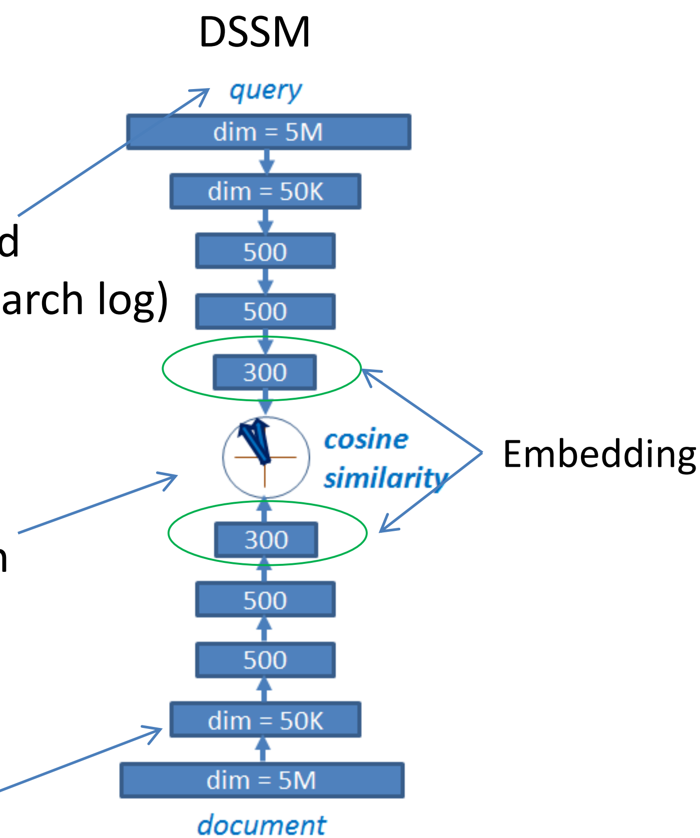
AE: reconstruction error
of the doc

DSSM: distance between
embedding vectors

Input:

AE: 1-hot word vector

DSSM: letter-trigram

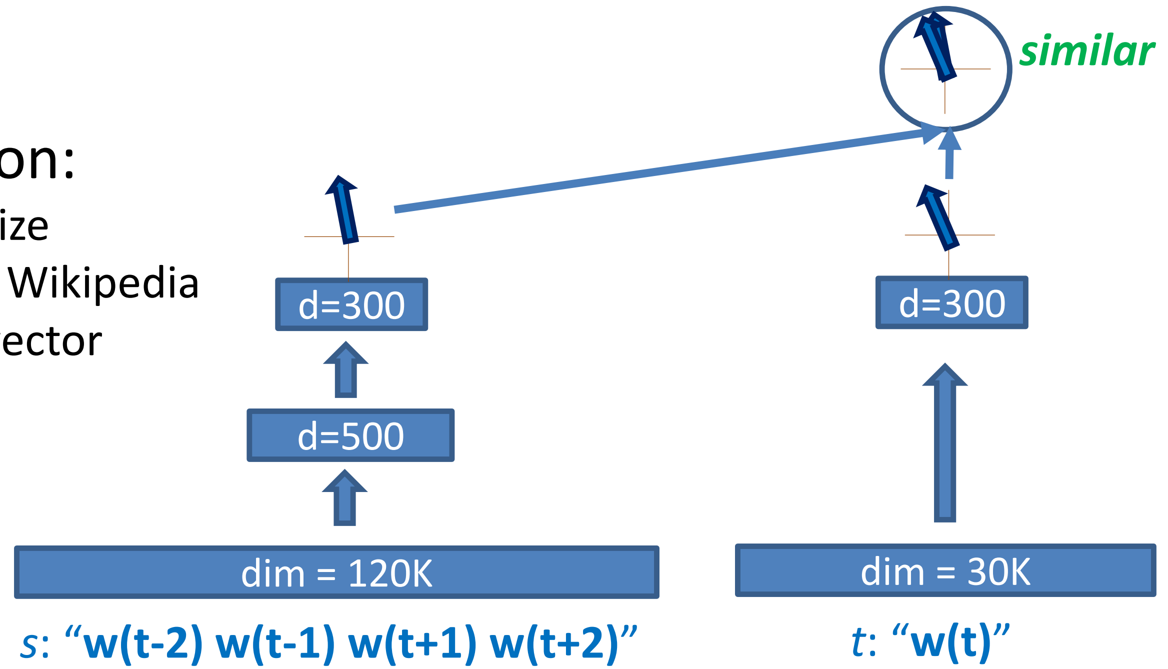


The DSSM can be trained using a variety of weak supervision signals without human labeling effort (e.g., user behavior log data).

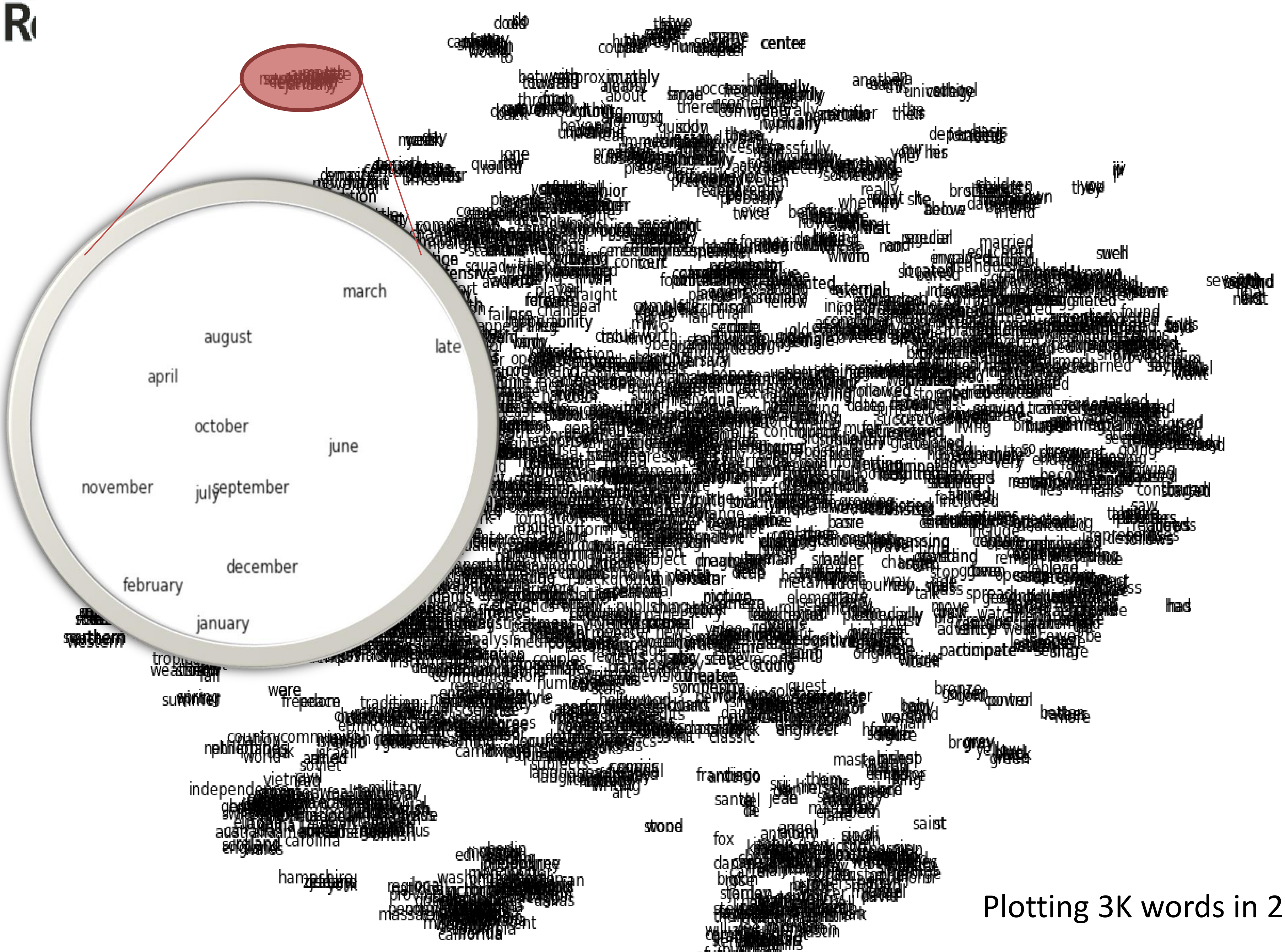
DSSM for Semantic Word Clustering and Analogy

- Learn word embedding by means of its neighbors (context)
  - Construct context <-> word training pair for DSSM
  - Similar words with similar context -> higher cosine

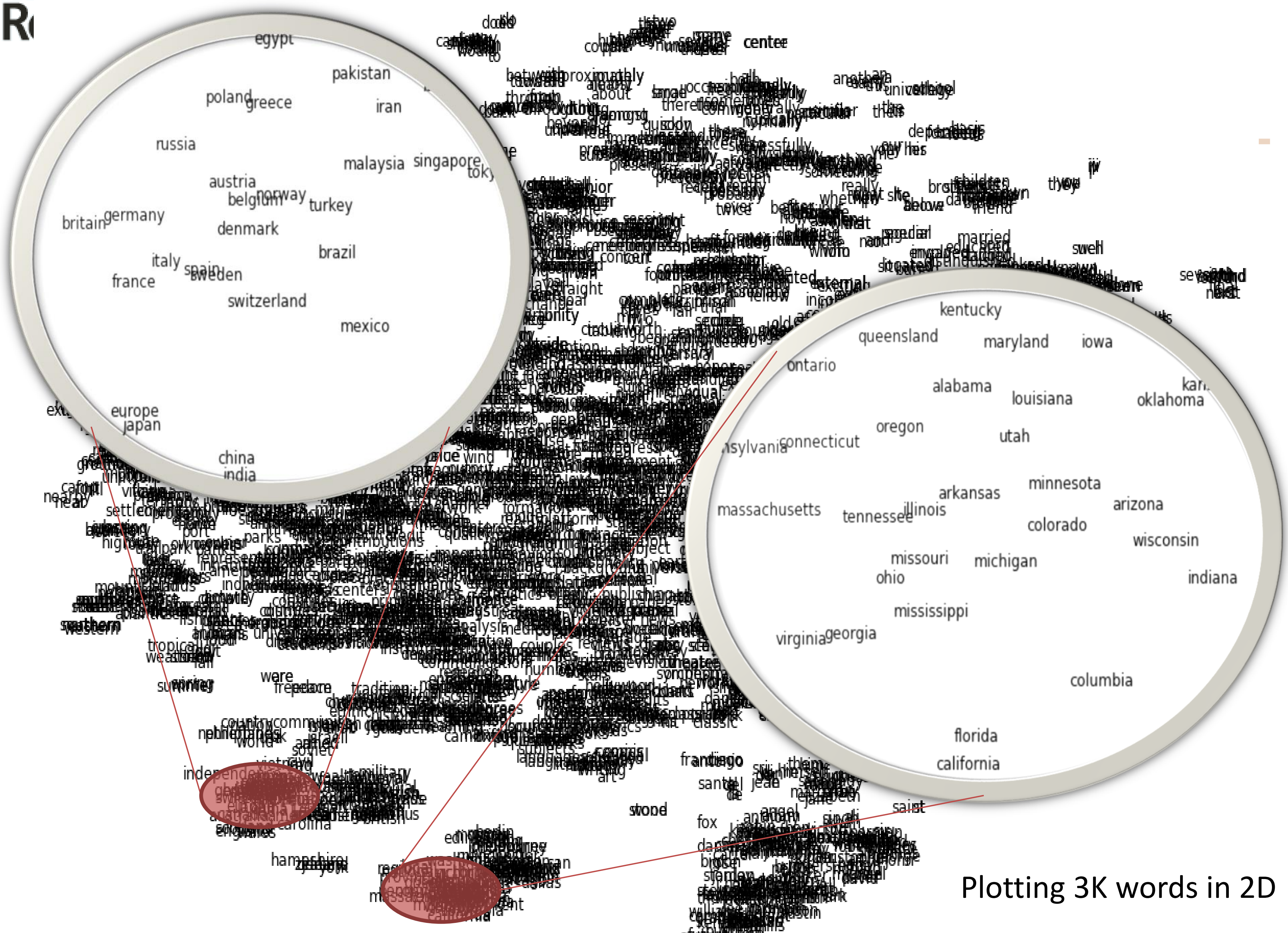
- Training Condition:
  - 30K vocabulary size
  - 10M words from Wikipedia
  - 50-dimentional vector



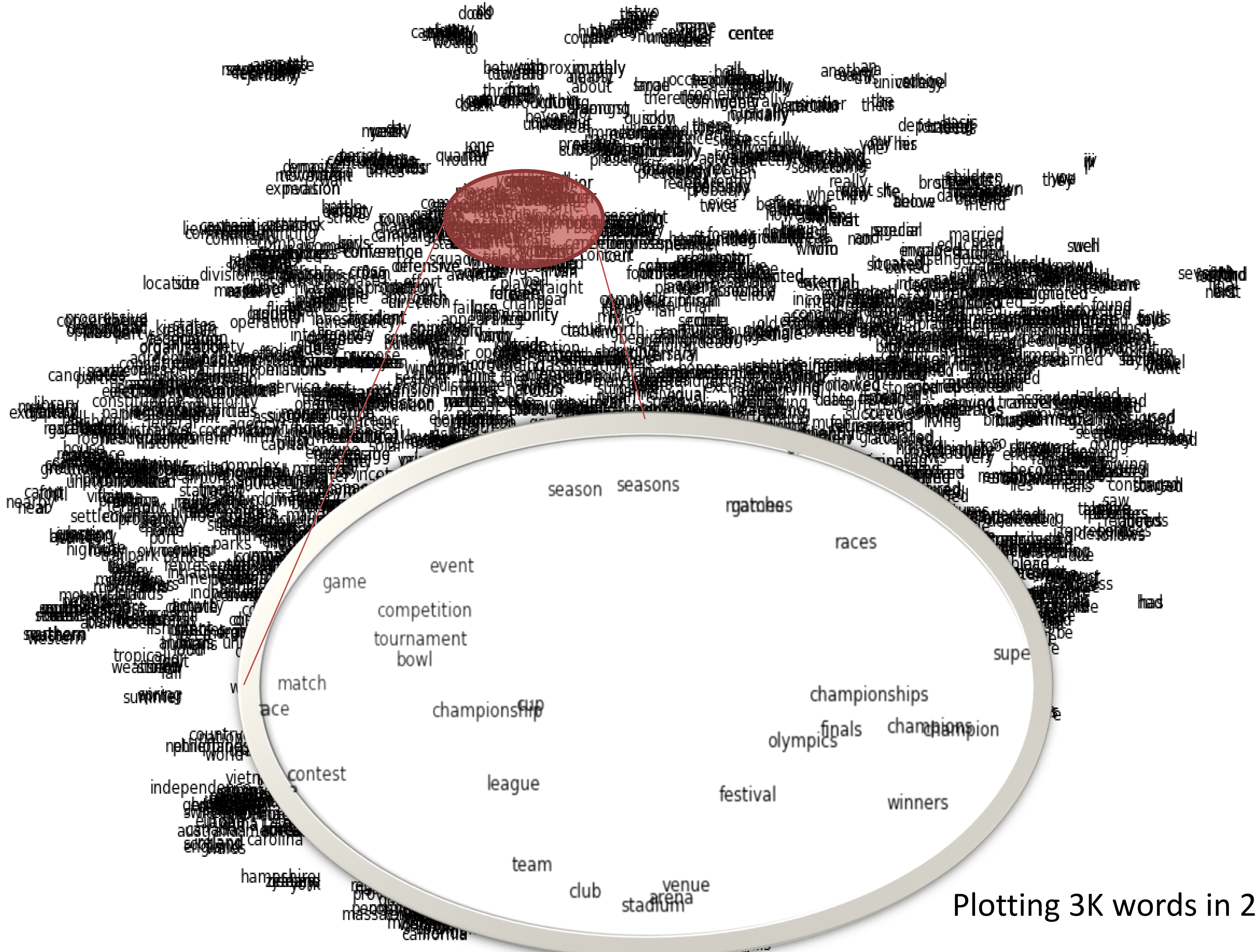
[Song et al. 2014]



Plotting 3K words in 2D



Plotting 3K words in 2D



Plotting 3K words in 2D

Microsoft Research

DSSM for Semantic Word Clustering and Analogy

Semantic clustering examples: top 3 neighbors of each word

king	earl (0.77)	pope (0.77)	lord (0.74)
woman	person (0.79)	girl (0.77)	man (0.76)
france	spain (0.94)	italy (0.93)	belgium (0.88)
rome	constantinople (0.81)	paris (0.79)	moscow (0.77)
winter	summer (0.83)	autumn (0.79)	spring (0.74)
rain	rainfall (0.76)	storm (0.73)	wet (0.72)
car	truck (0.8)	driver (0.73)	motorcycle (0.72)

Semantic analogy examples (following the task in Mikolov et al., 2013)

$$w_1 : w_2 = w_3 : ? \quad \Rightarrow \quad V_? = V_3 - V_1 + V_2$$

summer : rain = winter : ?	snow (0.79)	rainfall (0.73)	wet (0.71)
italy : rome = france : ?	paris (0.78)	constantinople (0.74)	egypt (0.73)
man : eye = car : ?	motor (0.64)	brake (0.58)	overhead (0.58)
man : woman = king : ?	mary (0.70)	prince (0.70)	queen (0.68)
read : book = listen : ?	sequel (0.65)	tale (0.63)	song (0.60)

[Song et al. 2014]

Microsoft Research

Interim Summary

- Word embedding
- Sub-word embedding gives a decomposable robust word representation
- The phrase/document level semantic embedding
- Using the DSSM to learn semantic embedding for phrases and documents

Part IV

Deep Learning in Machine Translation

Microsoft Research

Statistical machine translation (SMT)

C: 救援 人员 在 倒塌的 房屋 里 寻找 生还者
E: Rescue workers search for survivors in collapsed houses

Statistical decision: $E^* = \underset{E}{\operatorname{argmax}} P(E|C)$

Source-channel model: $E^* = \underset{E}{\operatorname{argmax}} P(C|E)P(E)$

Translation models: $P(C|E)$ and $P(E|C)$

Log-linear model: $P(E|C) = \frac{1}{Z(C,E)} \exp \sum_i \lambda_i h_i(C,E)$

Evaluation metric: BLEU score (higher is better)

[Koehn 2009]

Generative modeling for $P(E|C)$

- Story making (art)
  - how a target sentence is generated from a source sentence step by step
- Mathematical formulation (science)
  - modeling each generation step in the generative story using a probability distribution
- Parameter estimation (engineering)
  - implementing an effective way of estimating the probability distributions from training data

Translation modeling: $P(E|C)$

- Translation process (generative story)
  - *C* is broken into translation units
  - Each unit is translated into English
  - Glue translated units to form *E*
- Translation models
  - Word-based models
  - Phrase-based models
  - Syntax-based models

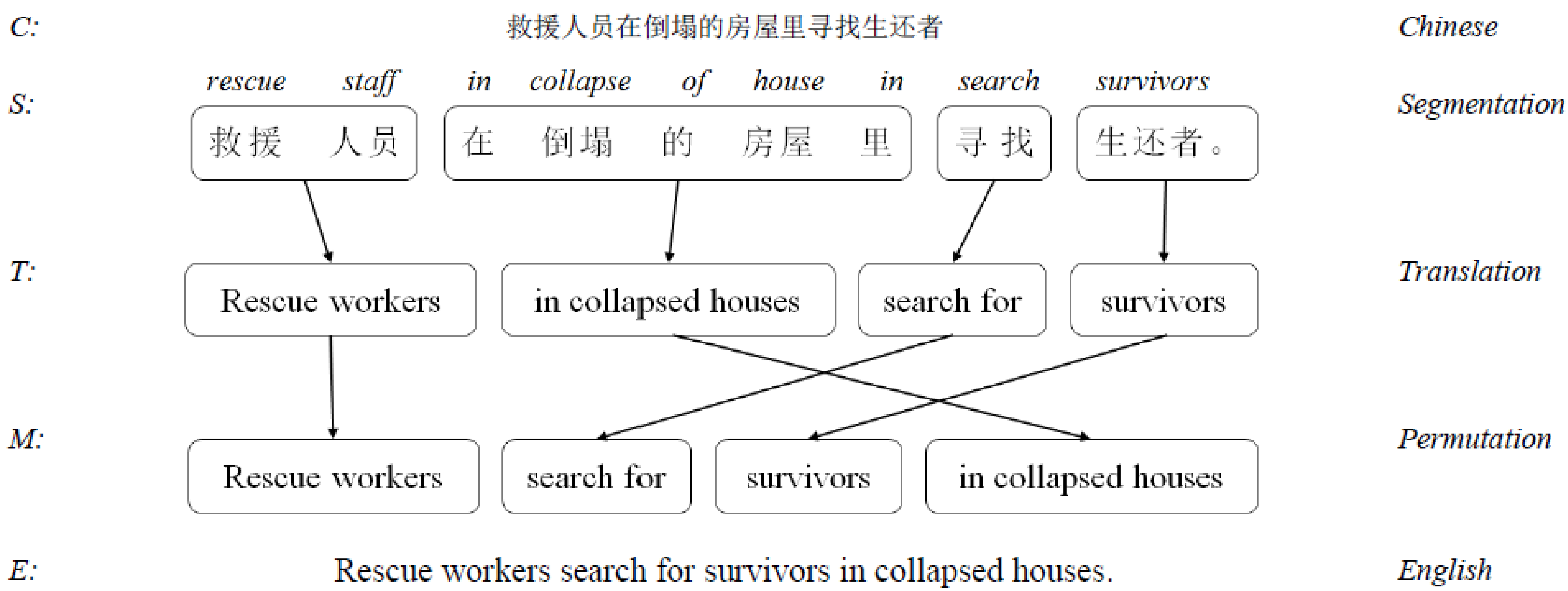
Microsoft Research

Phrase-based models

C: 救援人员在倒塌的房屋里寻找生还者 *Chinese*

Microsoft Research

Phrase-based models

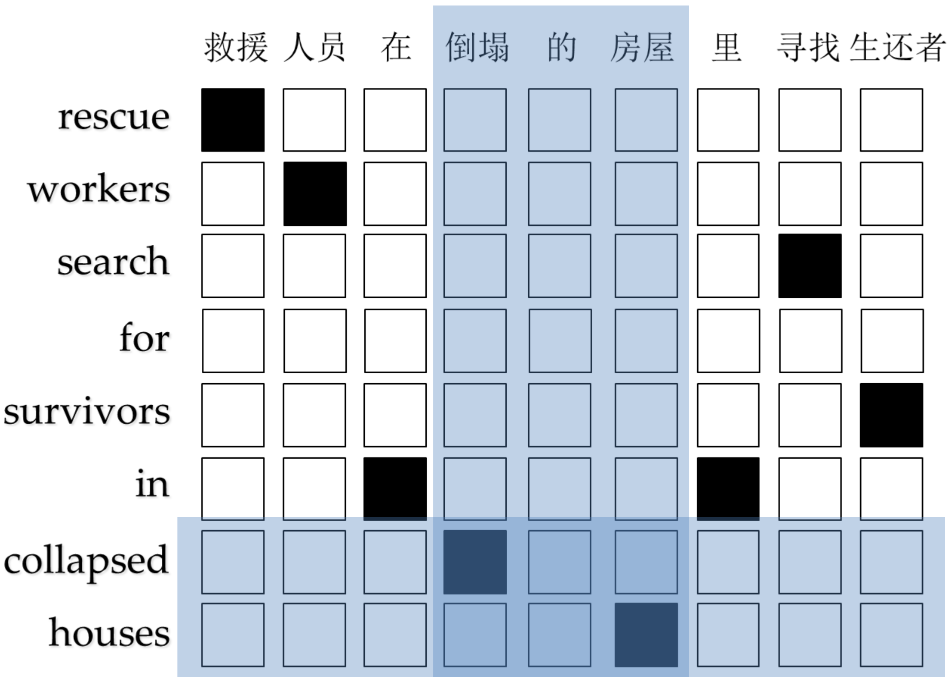


Microsoft Research

Mathematical formulation

- Assume a uniform probability over segmentations
  - $P(E|C) \propto \sum_{\substack{(S,T,M)\in \\ B(C,E)}} P(T|C,S) \cdot P(M|C,S,T)$
- Use the maximum approximation to the sum
  - $P(E|C) \approx \max_{\substack{(S,T,M)\in \\ B(C,E)}} P(T|C,S) \cdot P(M|C,S,T)$
- Assume each phrase being translated independently and use distance-based reordering model
  - $P(E|C) \propto \max_{\substack{(S,T,M)\in \\ B(C,Q)}} \prod_{k=1}^{K} P(\mathbf{e}_k|\mathbf{c}_k) d(start_i - end_{i-1} - 1)$

Parameter estimation

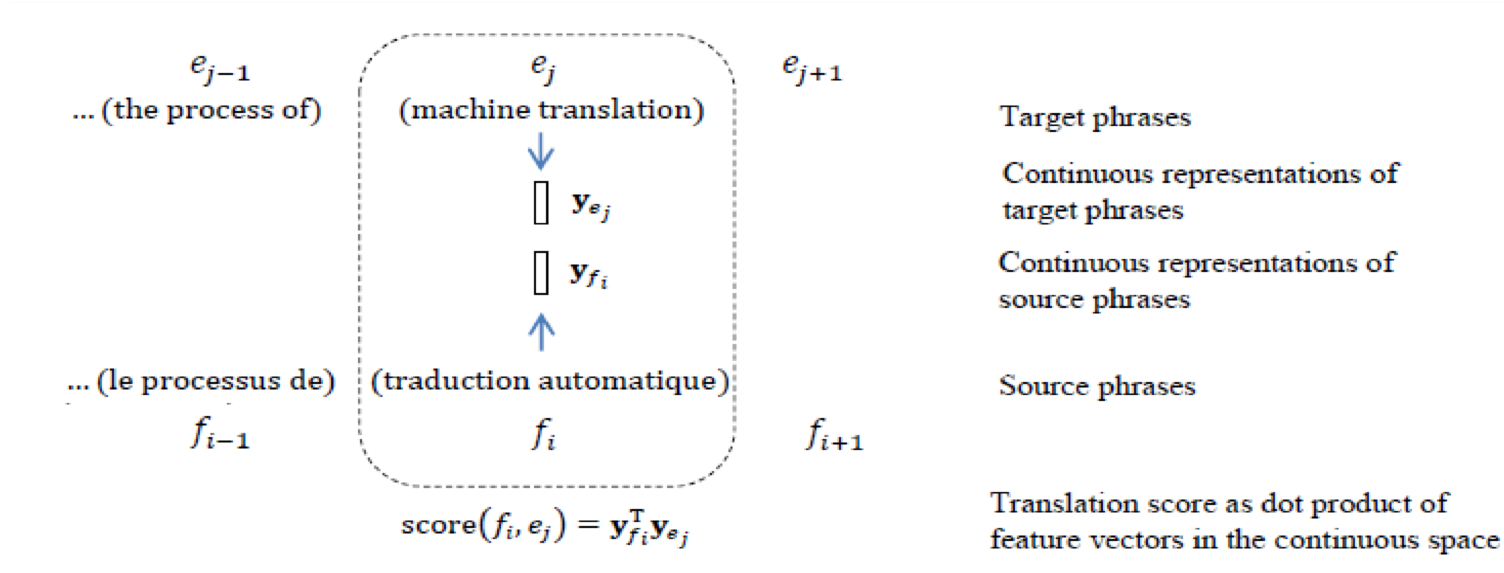


(救援, rescue)
(人员, workers)
(在, in)
(倒塌, collapsed)
(房屋, house)
(里, in)
(寻找, search)
(生还者, survivors)
(救援 人员, rescue workers)
(在 倒塌, in collapsed)
(倒塌 的, collapsed)
(的 房屋, house)
(寻找, search for)
(寻找 生还者, search for survivors)
(生还者, for survivors)
(倒塌 的 房屋, collapsed house)

MLE: $P(\mathbf{e}|\mathbf{c}) = \frac{N(\mathbf{c},\mathbf{e})}{\sum_{\mathbf{e}'} N(\mathbf{c},\mathbf{e}')}$

Don't forget smoothing

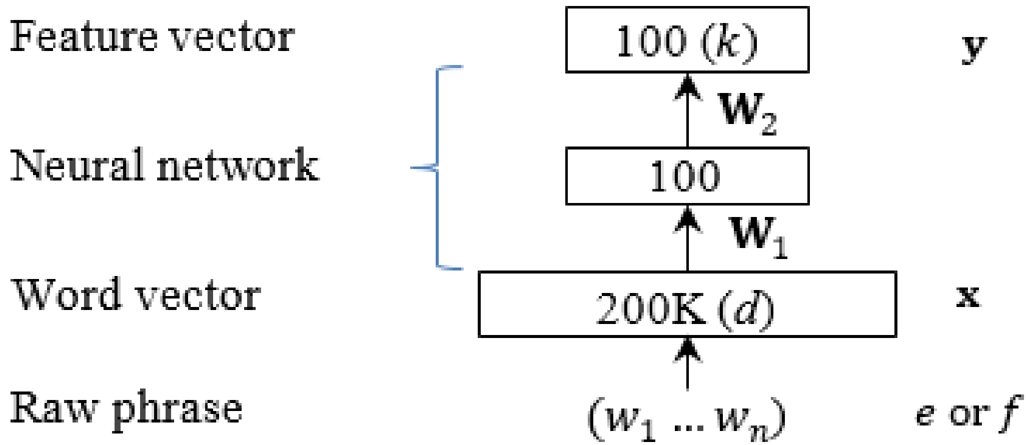
DSSM for phrase translation modeling



- Follows the "story" of phrase translation models, but
- Uses different parameter estimation method
  - Map source/target phrases into the same semantic space
  - Phrase translation score == similarity between their feature vectors in semantic space

[Gao, He, Yih, Deng, 2014]

A closer look at the *mapping*



- Bag-of-words representation of a phrase: $\mathbf{x}$
- Map $\mathbf{x}$ to a low-dim semantic space: $\phi(\mathbf{x})$: $\mathbb{R}^d \to \mathbb{R}^k$
- Mapping is performed using a neural net:

$$\mathbf{y} \equiv \phi(\mathbf{x}) = \tanh\left(\mathbf{W}_2{}^{\mathrm{T}}\left(\tanh\left(\mathbf{W}_1{}^{\mathrm{T}}\mathbf{x}\right)\right)\right)$$

- Translation score as similarity between feature vectors

$$\mathrm{score}(f, e) \equiv \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e) = \mathbf{y}_f^{\mathrm{T}}\mathbf{y}_e$$

[Gao, He, Yih, Deng, 2014]

Using the DSSM for SMT

- Define a new translation feature:

$$h_{M+1}(F_i, E, \boldsymbol{\theta}) = \sum_{(f,e)\in A} \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f, \mathbf{x}_e)$$

- Integrate into the log-linear model for SMT:

$$P(E|F) = \frac{1}{Z(F,E)} \exp \sum_i \lambda_i h_i(F,E)$$

$$E^* = \underset{E}{\operatorname{argmax}} \sum_i \lambda_i h_i(F,E)$$

[Gao, He, Yih, Deng, 2014]

Parameter estimation

- Parameters $(\boldsymbol{\lambda}, \boldsymbol{\theta})$
  - $\boldsymbol{\lambda}$: a handful of parameters in log-linear model.
  - $\boldsymbol{\theta}$: projection matrices of the DSSM.
- Take three steps to learn $(\boldsymbol{\lambda}, \boldsymbol{\theta})$:
  - Generate N-best lists using a baseline SMT system
  - Fix $\boldsymbol{\lambda}$, and optimize $\boldsymbol{\theta}$ w.r.t. a loss function on the N-best lists of training data.
  - Fix $\boldsymbol{\theta}$, and optimize $\boldsymbol{\lambda}$ to maximize BLEU on development data.

[Gao, He, Yih, Deng, 2014]

Training DSSM parameters, $\boldsymbol{\theta}$

- Define a loss function $\mathcal{L}(\boldsymbol{\theta})$, which is
  - Friendly to optimizer: differentiable/convex
  - Aiming the right target: closely related to task-specific metric (BLEU)
- Update $\boldsymbol{\theta}$ with gradient descent

$$\boldsymbol{\theta}^{new} = \boldsymbol{\theta} - \eta \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

- Algorithms
  - Batch training, L-BFGS
  - Stochastic Gradient Descent (SGD)

[Gao, He, Yih, Deng, 2014]

Microsoft Research

Loss function: $\mathcal{L}(\boldsymbol{\theta})$

- Expected BLEU based on n-best list
  - $\mathrm{xBleu}(\boldsymbol{\theta}) = \sum_{E \in \mathrm{GEN}(F_i)} P(E|F_i)\mathrm{sBleu}(E_i, E)$
  - $P(E|F_i) = \dfrac{\exp\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i,E,A)+\lambda_{M+1}h_{M+1}(F_i,E,\boldsymbol{\theta})\right)}{\sum_{E \in \mathrm{GEN}(F_i)} \exp(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(F_i,E,A)+\lambda_{M+1}h_{M+1}(F_i,E,\boldsymbol{\theta}))}$
- Friendly to optimizer?
  - Differentiable but non-convex
- Aiming the right target?
  - Closely related to BLEU

[Gao, He, Yih, Deng, 2014]

Microsoft Research

Gradient: $\partial\mathcal{L}(\boldsymbol{\theta})/\boldsymbol{\partial\theta}$

- $\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \sum_{(f,e)} \frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)} \frac{\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)}{\partial\boldsymbol{\theta}}$
- Error term: $-\partial\mathcal{L}(\boldsymbol{\theta})/\partial \mathrm{sim}_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)$
  - how the overall loss changes with the translation score of the phrase pair
- $\partial sim_{\boldsymbol{\theta}}(\mathbf{x}_f,\mathbf{x}_e)/\partial\boldsymbol{\theta}$ can be computed via Back Propagation (BP)

[Gao, He, Yih, Deng, 2014]

Microsoft Research

Evaluation

- Two Europarl translation tasks
  - English-to-French (EN-FR)
  - German-to-English (DE-EN)
- Baseline
  - A state-of-the-art phrase-based SMT system, i.e., Moses
- Evaluation metric
  - case insensitive BLEU score
  - 1 reference

Microsoft Research

Results

#	Systems	EN-FR		DE-EN	
		TEST1	**TEST2**	**TEST1**	**TEST2**
1	**Baseline**	33.04	33.06	26.10	26.07
2	**MRF**	33.73	33.91	26.91	26.81
3	**DSSM**	**34.03**	**34.39**	**27.21**	**27.03**
4	**Topic model**	33.08	33.15	26.08	26.11
5	**DPM**	33.10	33.29	26.25	26.23

- MRF: Markov Random Fields with xBleu (Gao and He 2013)
- DSSM: DSSM with xBleu
- Topic model: generative bilingual topic model (Gao et al. 2011)
- DPM: discriminative linear projection model (Gao et al. 2011)

Interim Summary

- Map the sentences in source/target languages into the same, language-independent semantic space
- The DSSM-based semantic translation model leads up to 1.3 BLEU improvement
- DSSM training: end2end optimization based on a task-specific objective
- Other DNNs for SMT
  - [Auli et al. 2013; Auli and Gao, 2014; Hu et al. 2014; Devlin et al. 2014]

Part V

Deep Learning of Semantic Similarity Models for Web Search and Beyond

Microsoft Research

Deep Structured Semantic Model (DSSM): learning semantic similarity between *X* and *Y*

Tasks	*X*	*Y*
Web search	***Search query***	***Web documents***
Ad selection	***Search query***	***Ad keywords***
Entity ranking	***Mention (highlighted)***	***Entities***
Recommendation	***Doc in reading***	***Interesting things in doc or other docs***
Machine translation	***Sentence in language A***	***Translations in language B***
Nature User Interface	*Command (text/speech)*	*Action*
Summarization	*Document*	*Summary*
Query rewriting	*Query*	*Rewrite*
Image retrieval	*Text string*	*Images*
…	…	…

[Huang et al. 2013; Shen et al. 2014; Gao et al. 2014a; Gao et al. 2014b]

An example of web search

Best Home Remedies for Cold and Flu

Wind Heat External Pathogens

By: Catherine Browne, L.Ac., MH, Dipl. Ac.

In Chinese medicine, colds and flu's are delineated into several different energetic classifications. Here we will outline the different types of cold and flu viruses that you will likely encounter, and then describe the best home remedies for these specific patterns that you can use to treat the cold or influenza virus.

Cold and Flu Basics

The basic pathogenic influences are:

- **Wind**
- **Cold**
- **Heat**
- **Damp**

Wind

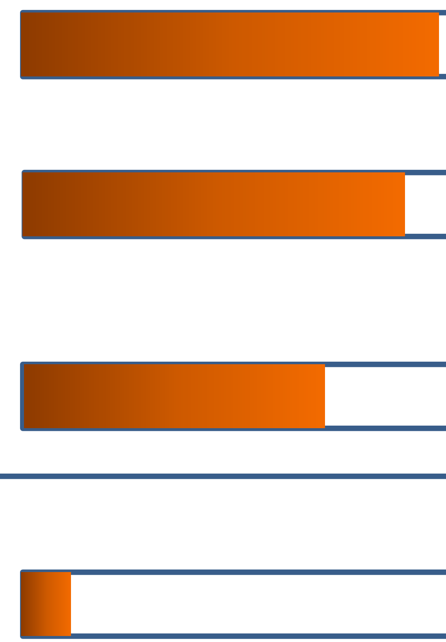
Theoretically, wind enters the body through the back of the neck area or nose carrying the pathogen. It first attacks the Lung system (including the sinuses) because the Lung organ system is the most external Yin organ, a thus the most vulnerable to an external invasion. External Wind invasion is marked by acute conditions with a sudden onset of symptoms.

- cold home remedy
- cold remeedy
- flu treatment
- how to deal with stuffy nose

Smart matching between Q and D

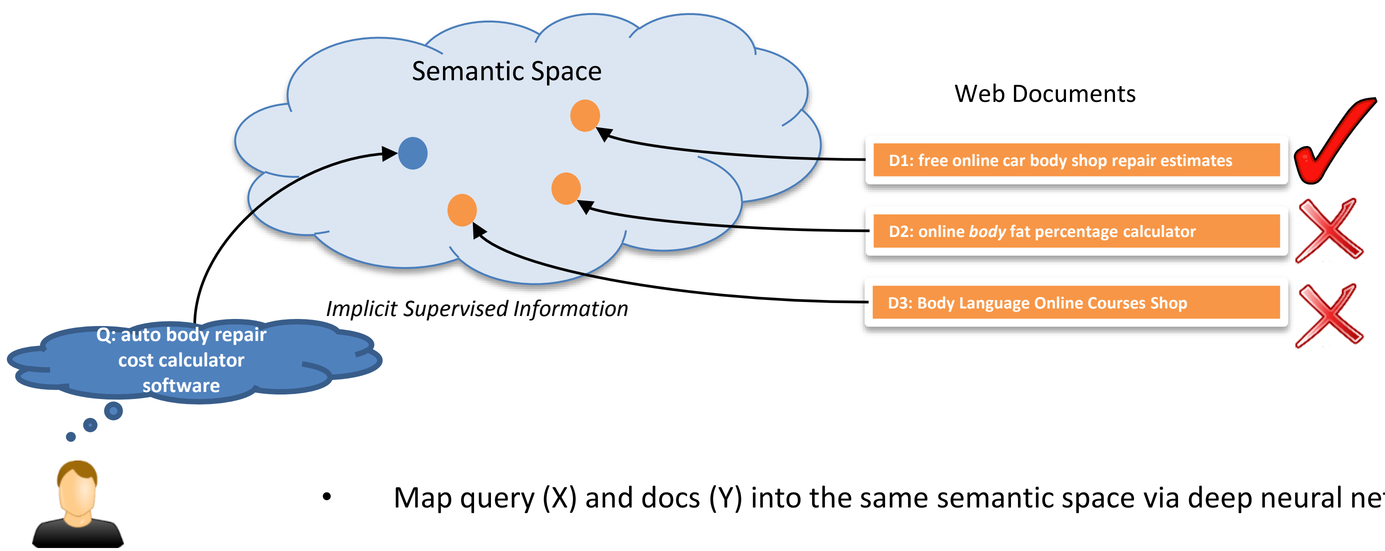
- Fuzzy keyword matching
  - Q: cold home remedy
  - D: best home remedies for cold and flu
- Spelling correction
  - Q: cold remeedies
  - D: best home remedies for cold and flu
- Query alteration/expansion
  - Q: flu treatment
  - D: best home remedies for cold and flu
- **Query/document semantic matching**
  - Q: how to deal with stuffy nose
  - D: best home remedies for cold and flu
  - Q: auto body repair cost calculator software
  - D: free online car body shop repair estimates

R&D progress



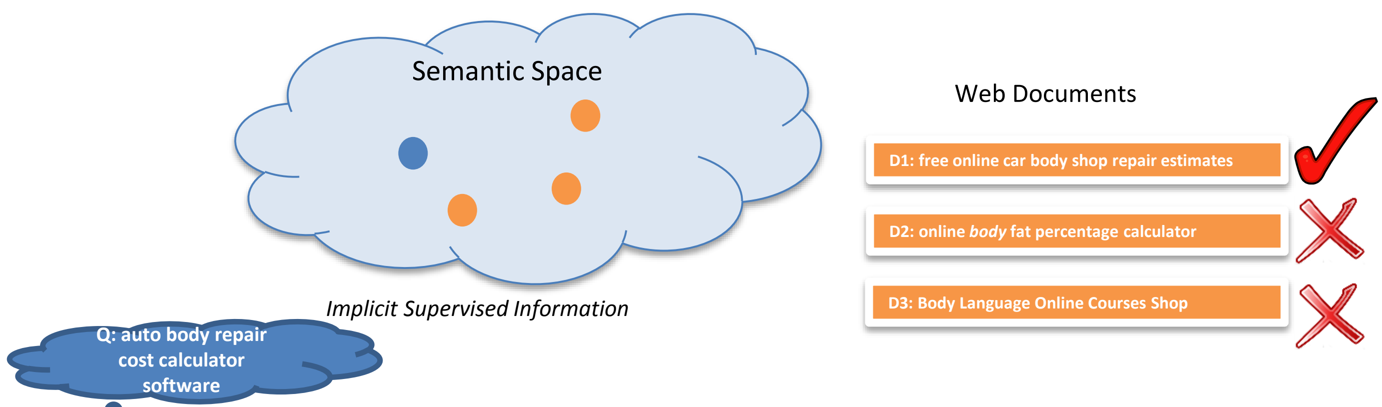
Microsoft Research

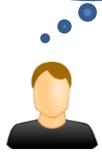
Learning DSSM on labeled X-Y pairs (clicked Q-D pairs)



- Map query (X) and docs (Y) into the same semantic space via deep neural net

Learning DSSM on labeled X-Y pairs (clicked Q-D pairs)

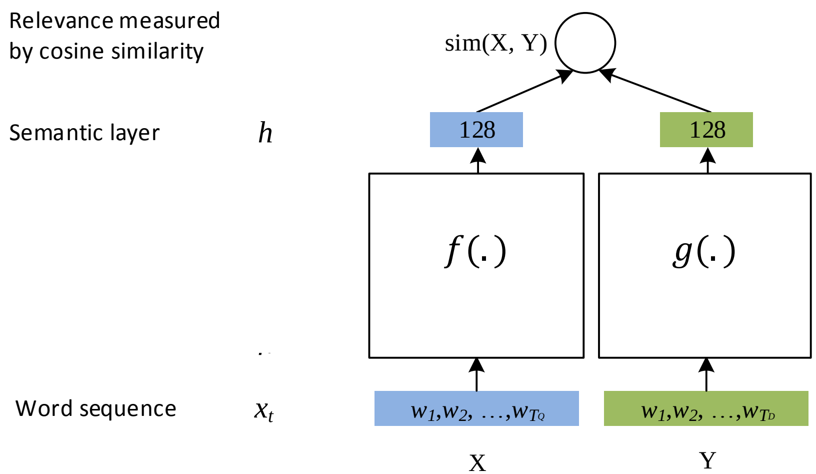




- Map query (X) and docs (Y) into the same semantic space via deep neural net
- Relevant docs are closer to query than irrelevant docs in that space

Microsoft Research

DSSM: explore the power of deep learning



Learning: maximize the similarity between relevant queries and docs

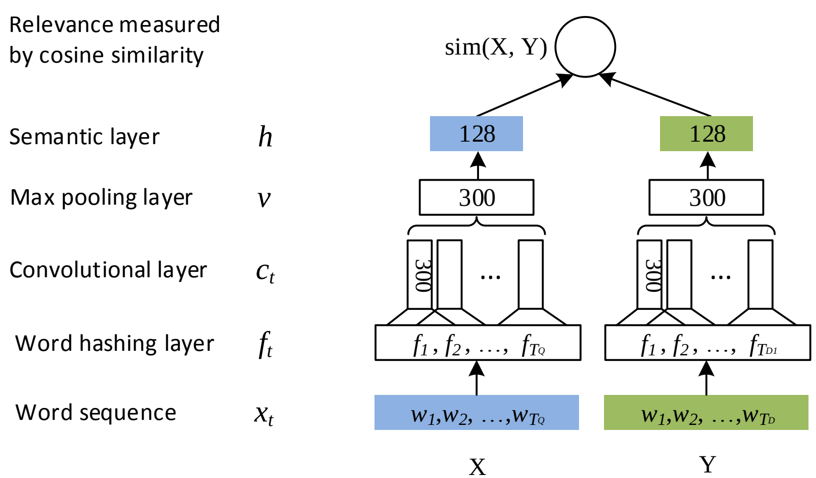
Representation: use DNN to extract abstract semantic representations

DSSM combines three pieces of MSR work

- DNN structure follows deep auto-encoder (Deng, Seltzer, Hinton, et al. 2010)
- The use of search logs for translation model training (Gao, He, and Nie, 2010)
- Parameter optimization uses the pairwise rank loss based on cosine similarity (Yih et al. 2011; Gao et al. 2011)

[Shen, He, Gao, Deng, Mesnil, 2014]

DSSM: explore the power of deep learning



Learning: maximize the similarity between relevant queries and docs

Representation: use DNN to extract abstract semantic representations

Convolutional and Max-pooling layer: identify key words/concepts in Q and D

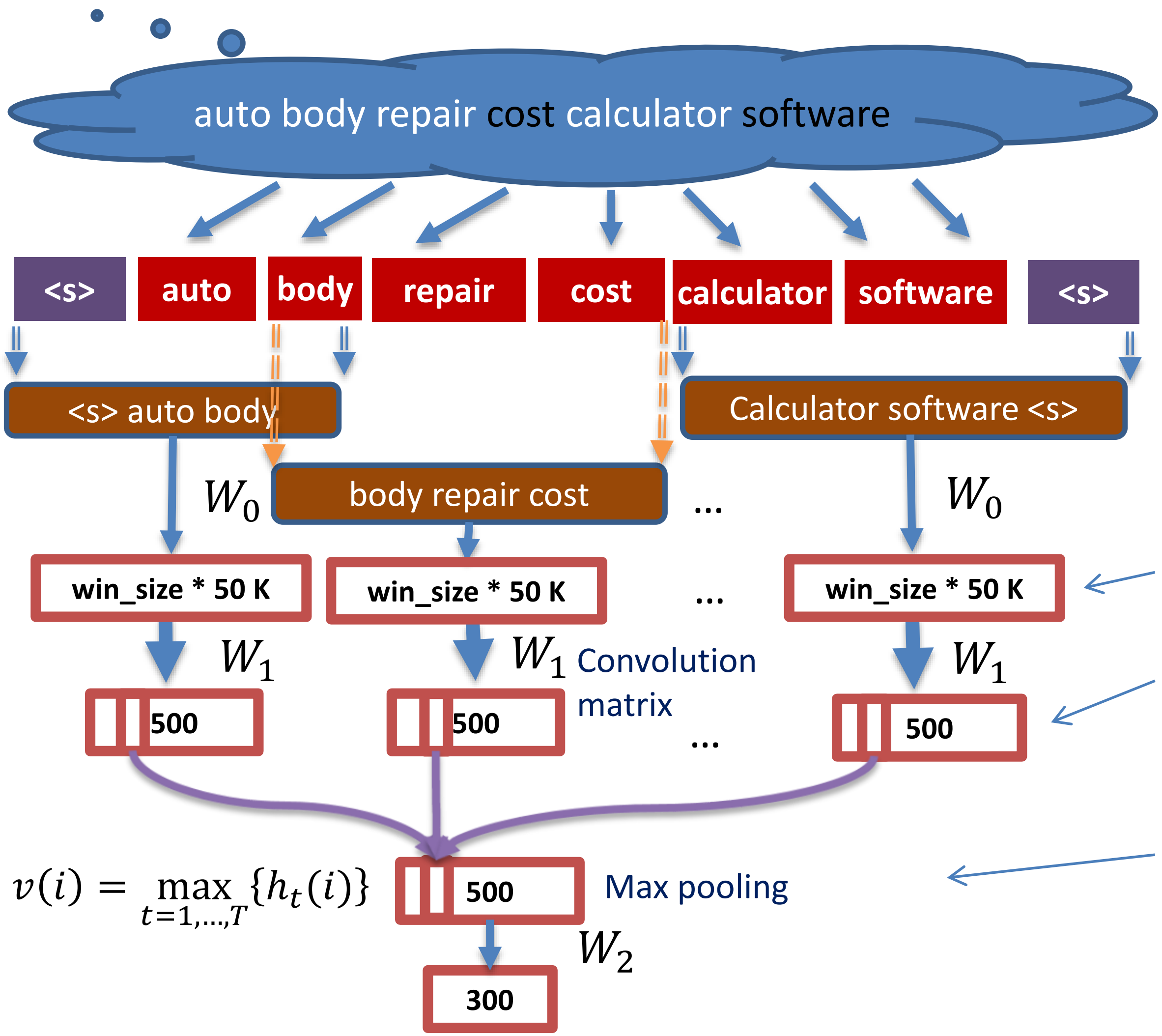
Word hashing: use sub-word unit (e.g., letter-ngram) as raw input to handle very large vocabulary

DSSM combines three pieces of MSR work

- DNN structure follows deep auto-encoder (Deng, Seltzer, Hinton, et al. 2010)
- The use of search logs for translation model training (Gao, He, and Nie, 2010)
- Parameter optimization uses the pairwise rank loss based on cosine similarity (Yih et al. 2011; Gao et al. 2011)

[Shen, He, Gao, Deng, Mesnil, 2014]

Example: search intent identification



Query as a word sequence rather than "bag of words"

Sliding Window input:
n-gram phrase (n = 3)

Letter-trigram representation

Convolutional Layer h: generate word-within-context embedding

Max Pooling Layer v: identify key words in a query

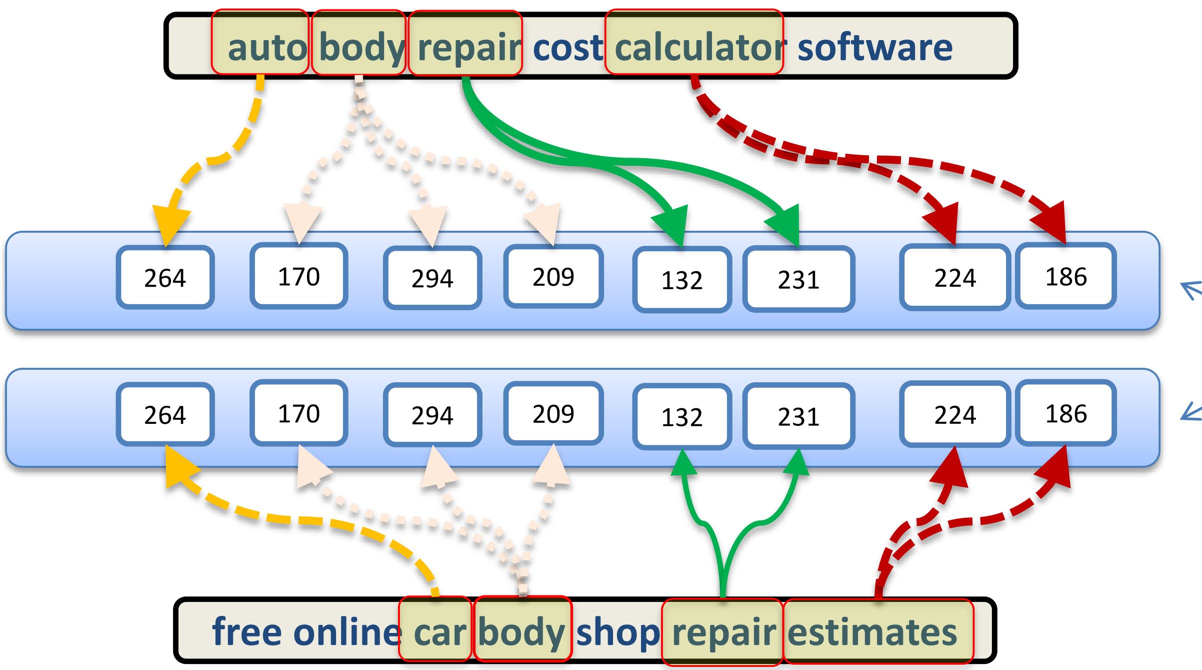
Semantic Layer y

Convolutional and max-pooling layers

- Extract local features using convolutional layer
  - {w2, w3} → topic blue
  - {w5, w6} → topic green
- Generate global features using max-pooling
  - Key topics of the doc → blue and green
  - keywords of the doc: w2-w3 and w5-w6
  - Link btw keywords and key topics

Intent matching via convolutional DSSM

- Semantic matching of query and document



Most active neurons at the **max-pooling layers** of the query and document nets, respectively

Microsoft Research

More examples

Query	Title of the top-1 returned document retrieved by CLSM
warm environment arterioles do what	**thermoregulation** wikipedia the free encyclopedia
auto body repair cost **calculator** software	free online **car body shop repair estimates**
what happens if our body **absorbs excessive** amount **vitamin d**	**calcium** supplements and **vitamin d** discussion stop **sarcoidosis**
how do **camera** use **ultrasound focus** automatically	**wikianswers how** does a **camera focus**
how to **change font excel office 2013**	**change font** default styles in **excel 2013**
where do i get my **federal tax return transcript**	how to get **trasncripts** of **federal** income **tax returns** fast ehow
12 **fishing boats trailers**	**trailer kits** and accessories **motorcycle** utility **boat snowmobile**
acp **ariakon combat pistol** 2.0	paintball acp **combat** pistol paintball **gun paintball pistol** package deal marker and gun

Training C-DSSM from Query-Doc pairs

- Mini-batch SGD on GPU
- Objective: **Bayes Risk** based on cosine similarity
- For each query Q, there is a set of documents $\boldsymbol{D}$
  - $\boldsymbol{D}$ can be constructed via sampling
  - Each D in $\boldsymbol{D}$ has a relevance label w.r.t. Q
- $P(D|Q) = \frac{\exp(\gamma R(Q,D))}{\sum_{D\prime\in\boldsymbol{D}} \exp(\gamma R(Q,D\prime))}$,
  - $R(Q,D)$ is cosine similarity
- $\text{loss}(Q,\boldsymbol{D}) = \sum_{D\in\boldsymbol{D}} P(D|Q)\text{cost}(Q,D)$,
  - $\text{cost}(.)$ is a function of relevance label

Microsoft Research

Mine Q-D pairs from search logs

how to deal with stuffy nose?	⟺	NO CLICK
stuffy nose treatment	⟺	NO CLICK
cold home remedies	⟺	http://www.agelessherbs.com/BestHomeRemediesColdFlu.html

[Gao, He, Nie, 2010]

Microsoft Research

Mine Q-D pairs from search logs

how to deal with stuffy nose? ⟺

stuffy nose treatment ⟺

cold home remedies ⟺

Best Home Remedies for Cold and Flu

Wind Heat External Pathogens

By: Catherine Browne, L.Ac., MH, Dipl. Ac.

In Chinese medicine, colds and flu's are delineated into several different energetic classifications. Here we will outline the different types of cold and flu viruses that you will likely encounter, and then describe the best home remedies for these specific patterns that you can use to treat the cold or influenza virus.

Cold and Flu Basics

The basic pathogenic influences are:

- **Wind**
- **Cold**
- **Heat**
- **Damp**

Wind

Theoretically, wind enters the body through the back of the neck area or nose carrying the pathogen. It first attacks the Lung system (including the sinuses) because the Lung organ system is the most external Yin organ, a thus the most vulnerable to an external invasion. External Wind invasion is marked by acute conditions with a sudden onset of symptoms.

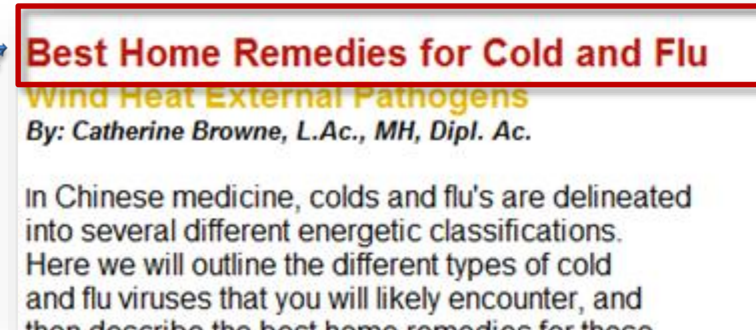
[Gao, He, Nie, 2010]

Mine Q-D pairs from search logs

how to deal with stuffy nose?

stuffy nose treatment

cold home remedies



QUERY (Q)	Title (T)
how to deal with stuffy nose	best home remedies for cold and flu
stuffy nose treatment	best home remedies for cold and flu
cold home remedies	best home remedies for cold and flu
… …	… …
go israel	forums goisrael community
skate at wholesale at pr	wholesale skates southeastern skate supply
breastfeeding nursing blister baby	clogged milk ducts babycenter
thank you teacher song	lyrics for teaching educational children s music
immigration canada lacolle	cbsa office detailed information

[Gao, He, Nie, 2010]

Evaluation Methodology

- Measurement: NDCG, t-test
- Test set:
  - 12,071 English queries sampled from 1-y log
  - 5-level relevance label for each query-doc pair
- Training data for translation models:
  - 82,834,648 query-title pairs
- Baselines
  - Lexicon matching models: BM25, ULM
  - Translation models
  - Topic models

Translation models for web search

D: best home remedies for cold and flu

Q: how to deal with stuffy nose

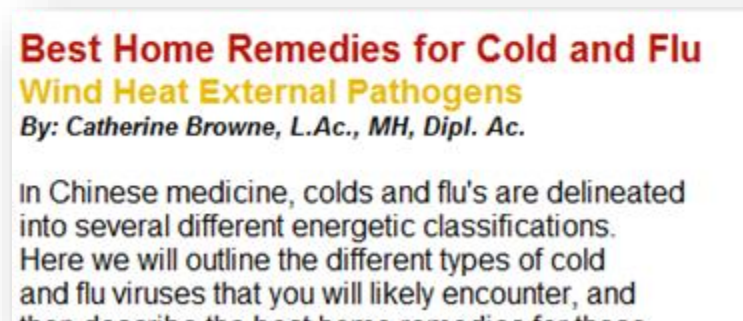
- Model documents and queries as different languages
- Cast mapping queries to documents as bridging the language gap via translation
- Leverage statistical machine translation (SMT) technologies and infrastructures to improve search relevance

[Gao, He, Nie, 2010]

SMT for document ranking

- Given a Q, D can be ranked by how likely it is that Q is "translated" from D, $P(\mathrm{Q}|\mathrm{D})$

how to deal with stuffy nose?


Best Home Remedies for Cold and Flu
Wind Heat External Pathogens
By: Catherine Browne, L.Ac., MH, Dipl. Ac.

In Chinese medicine, colds and flu's are delineated into several different energetic classifications. Here we will outline the different types of cold and flu viruses that you will likely encounter, and

- Word based models
- Phrase based models

Microsoft Research

Word based models

Sample IBM-1 word translation probability after EM training on the query-title pairs

q	$P(q\|w)$	Q	$P(q\|w)$
titanic	0.56218	Vista	0.80575
ship	0.01383	Windows	0.05344
movie	0.01222	Download	0.00728
pictures	0.01211	ultimate	0.00571
sink	0.00697	xp	0.00355
facts	0.00689	microsoft	0.00342
photos	0.00533	bit	0.00286
rose	0.00447	compatible	0.00270
people	0.00441	premium	0.00244
survivors	0.00369	free	0.00211

w = titanic w = vista

q	$P(q\|w)$	q	$P(q\|w)$
everest	0.52826	pontiff	0.17288
mt	0.02672	pope	0.09831
mount	0.02117	playground	0.03729
deaths	0.00958	wally	0.03053
person	0.00598	bartlett	0.03051
summit	0.00503	current	0.02712
climbing	0.00454	quantum	0.02373
cost	0.00446	wayne	0.02372
visit	0.00441	john	0.02034
height	0.00397	stewart	0.02031

w = everest w = pontiff

Microsoft Research

Phrase based models

q	$P(\mathbf{q}\|\mathbf{w})$	q	$P(\mathbf{q}\|\mathbf{w})$
titanic	0.43195	sierra vista	0.61717
rms titanic	0.03793	sv	0.02260
titanic sank	0.02114	vista	0.01678
titanic sinking	0.01695	sierra	0.01581
titanic survivors	0.01537	az	0.00417
titanic ship	0.01112	bella vista	0.00320
titanic sunk	0.00960	arizona	0.00223
titanic pictures	0.00593	dominoes sierra vista	0.00221
titanic exhibit	0.00540	dominos sierra vista	0.00221
ship titanic	0.00383	meadows	0.00029

w = rms titanic   **w** = sierra vista

Figure 6: Sample phrase translation probabilities learned from the word-aligned query-title pairs.

- Phrases, with context information, lead to less ambiguous translations than words

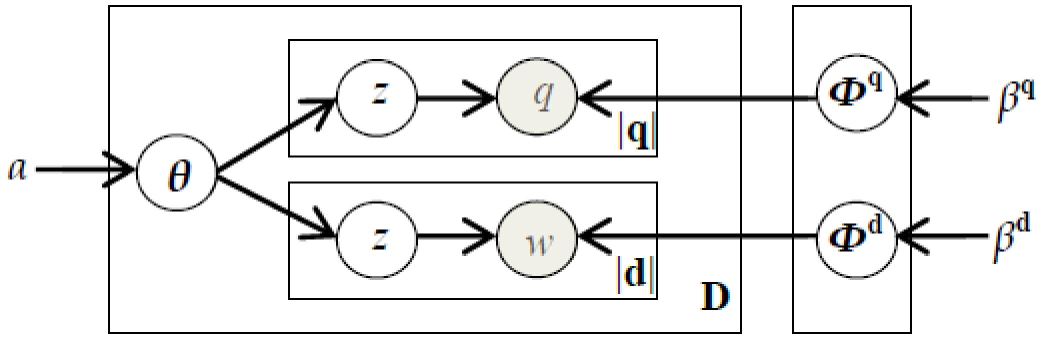
Generative Topic Models

Q: stuffy nose treatment ← D: cold home remedies

Q: stuffy nose treatment ← Topic ← D: cold home remedies

- Probabilistic latent Semantic Analysis (PLSA)
  - $P(\mathrm{Q}|\mathrm{D}) = \prod_{q \in \mathrm{Q}} \sum_{z} P(q|\boldsymbol{\phi}_z) P(z|\mathrm{D}, \boldsymbol{\theta})$
  - D is assigned a single most likely topic vector
  - Q is generated from the topic vectors
- Latent Dirichlet Allocation (LDA) generalizes PLSA
  - a posterior distribution over topic vectors is used
  - PLSA = LDA with MAP inference

Bilingual topic model for web search



- For each topic *z*: $(\boldsymbol{\phi}_z^{\mathrm{Q}}, \boldsymbol{\phi}_z^{\mathrm{D}}) \sim \mathrm{Dir}(\boldsymbol{\beta})$
- For each Q-D pair: $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$
- Each *q* is generated by $z \sim \boldsymbol{\theta}$ and $q \sim \boldsymbol{\phi}_z^{\mathrm{Q}}$
- Each *w* is generated by $z \sim \boldsymbol{\theta}$ and $w \sim \boldsymbol{\phi}_z^{\mathrm{D}}$

[Gao, Toutanova, Yih, 2011]

MAP Estimation via EM

- Estimate $(\boldsymbol{\theta}, \boldsymbol{\phi}^{\mathrm{Q}}, \boldsymbol{\phi}^{\mathrm{D}})$ by maximizing joint log likelihood of Q-D pairs and the parameters
- E-Step: compute posterior probabilities
  - $P(z|q, \boldsymbol{\theta}^{\mathrm{Q,D}})$,$P(z|w, \boldsymbol{\theta}^{\mathrm{Q,D}})$
- M-Step: update parameters using the posterior probabilities
  - $P(q|\boldsymbol{\phi}_z^{\mathrm{Q}})$,$P(w|\boldsymbol{\phi}_z^{\mathrm{D}})$, $P(z|\boldsymbol{\theta}^{\mathrm{Q,D}})$

Microsoft Research

Results

#	Models	NDCG@1	NDCG@3
	Lexical Matching Models		
1	BM25	30.5	32.8
2	Unigram LM	30.4 (-0.1)	32.7 (-0.1)
	Topic Models		
3	PLSA [Hofmann 1999]	30.5 (+0.0)	33.5 (+0.7)
4	BLTM [Gao et al. 2011]	31.6 (+1.1)	34.4 (+1.6)
	Clickthrough-based Translation Models		
5	WTM [Gao et al. 2010]	31.5 (+1.0)	34.2 (+1.4)
6	PTM [Gao et al. 2010]	31.9 (+1.4)	34.7 (+1.9)
	Deep Structure Semantic Model		
7	DSSM [Huang et al. 2013]	32.0 (+1.5)	35.5 (+2.7)
8	**C-DSSM [Shen et al. 2014]**	**34.2 (+3.7)**	**37.4 (+4.6)**

- Convolutional DSSM is the new state-of-the-art

[Shen, He, Gao, Deng, Mesnil, 2014]

Microsoft Research

Deep Structured Semantic Model (DSSM): learning semantic similarity between *X* and *Y*

Tasks	*X*	*Y*
Web search	***Search query***	***Web documents***
Ad selection	***Search query***	***Ad keywords***
Entity ranking	***Mention (highlighted)***	***Entities***
Recommendation	***Doc in reading***	***Interesting things in doc or other docs***
Machine translation	***Sentence in language A***	***Translations in language B***
Nature User Interface	*Command (text/speech)*	*Action*
Summarization	*Document*	*Summary*
Query rewriting	*Query*	*Rewrite*
Image retrieval	*Text string*	*Images*
…	…	…

[Huang et al. 2013; Shen et al. 2014; Gao et al. 2014a; Gao et al. 2014b]

Microsoft Research

Thank You

Microsoft Research

References

- Auli, M., Galley, M., Quirk, C. and Zweig, G., 2013. Joint language and translation modeling with recurrent neural networks. In EMNLP.
- Auli, M., and Gao, J., 2014. Decoder integration and expected bleu training for recurrent neural network language models. In ACL.
- Bengio, Y., 2009. Learning deep architectures for AI. Foundumental Trends Machine Learning, vol. 2.
- Bengio, Y., Courville, A., and Vincent, P. 2013. Representation learning: A review and new perspectives. IEEE Trans. PAMI, vol. 38, pp. 1798-1828.
- Bengio, Y., Ducharme, R., and Vincent, P., 2000. A Neural Probabilistic Language Model, in NIPS.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural language processing (almost) from scratch. in JMLR, vol. 12.
- Dahl, G., Yu, D., Deng, L., and Acero, 2012. A. Context-dependent, pre-trained deep neural networks for large vocabulary speech recognition, IEEE Trans. Audio, Speech, & Language Proc., Vol. 20 (1), pp. 30-42.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. J. American Society for Information Science, 41(6): 391-407
- Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., and Hinton, G., 2010. Binary Coding of Speech Spectrograms Using a Deep Auto-encoder, in Interspeech.
- Deng, L., Tur, G, He, X, and Hakkani-Tur, D. 2012. Use of kernel deep convex networks and end-to-end learning for spoken language understanding, Proc. IEEE Workshop on Spoken Language Technologies.
- Deng, L., Yu, D. and Acero, A. 2006. Structured speech modeling, IEEE Trans. on Audio, Speech and Language Processing, vol. 14, no. 5, pp. 1492-1504.
- Deng, L., Yu, D., and Platt, J. 2012. Scalable stacking and learning for building deep architectures, Proc. ICASSP.
- Deoras, A., and Sarikaya, R., 2013. Deep belief network based semantic taggers for spoken language understanding, in INTERSPEECH.
- Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J., 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation, ACL.
- Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T., 2013. DeViSE: A Deep Visual-Semantic Embedding Model, Proc. NIPS.
- Gao, J., He, X., Yih, W-t., and Deng, L. 2014a. Learning continuous phrase representations for translation modeling. In ACL.
- Gao, J., He, X., and Nie, J-Y. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In CIKM.
- Gao, J., Pantel, P., Gamon, M., He, X., and Deng, L. 2014b. Modeling interestingness with deep neural networks. MSR Tech Report.
- Gao, J., Toutanova, K., Yih., W-T. 2011. Clickthrough-based latent semantic models for web search. In SIGIR.
- Gao, J., Yuan, W., Li, X., Deng, K., and Nie, J-Y. 2009. Smoothing clickthrough data for web search ranking. In SIGIR.
- Gao, J., and He, X. 2013. Training MRF-based translation models using gradient ascent. In NAACL-HLT.
- Graves, A., Jaitly, N., and Mohamed, A., 2013a. Hybrid speech recognition with deep bidirectional LSTM, Proc. ASRU.
- Graves, A., Mohamed, A., and Hinton, G., 2013. Speech recognition with deep recurrent neural networks, Proc. ICASSP.

Microsoft Research

References

- He, X. and Deng, L., 2013. Speech-Centric Information Processing: An Optimization-Oriented Approach, in Proceedings of the IEEE.
- He, X. and Deng, L., 2012. Maximum Expected BLEU Training of Phrase and Lexicon Translation Models , ACL.
- He, X., Deng, L., and Chou, W., 2008. Discriminative learning in sequential pattern recognition, Sept. IEEE Sig. Proc. Mag.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97.
- Hinton, G., and Salakhutdinov, R., 2010. Discovering binary codes for documents by learning deep generative models. Topics in Cognitive Science.
- Hu, Y., Auli, M., Gao, Q., and Gao, J. 2014. Minimum translation modeling with recurrent neural networks. In EACL.
- Huang, E., Socher, R., Manning, C, and Ng, A. 2012. Improving word representations via global context and multiple word prototypes, Proc. ACL.
- Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In CIKM.
- Hutchinson, B., Deng, L., and Yu, D., 2012. A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition, Proc. ICASSP.
- Hutchinson, B., Deng, L., and Yu, D., 2013. Tensor deep stacking networks, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, pp. 1944 - 1957.
- Kiros, R., Zemel, R., and Salakhutdinov, R. 2013. Multimodal Neural Language Models, Proc. NIPS Deep Learning Workshop.
- Koehn, P. 2009. Statistical Machine Translation. Cambridge University Press.
- Krizhevsky, A., Sutskever, I, and Hinton, G., 2012. ImageNet Classification with Deep Convolutional Neural Networks, NIPS.
- Le, H-S, Oparin, I., Allauzen, A., Gauvain, J-L., Yvon, F., 2013. Structured output layer neural network language models for speech recognition, IEEE Transactions on Audio, Speech and Language Processing.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition, Proceedings of the IEEE, Vol. 86, pp. 2278-2324.
- Li, P., Hastie, T., and Church, K.. 2006. Very sparse random projections, in Proc. SIGKDD.
- Mesnil, G., He, X., Deng, L., and Bengio, Y., 2013. Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding, in Interspeech.
- Mikolov, T. 2012. Statistical Language Models based on Neural Networks, Ph.D. thesis, Brno University of Technology.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space, Proc. ICLR.
- Mikolov, T., Kombrink,. S., Burget, L., Cernocky, J., Khudanpur, S., 2011. Extensions of Recurrent Neural Network LM. ICASSP.
- Mikolov, T., Yih, W., Zweig, G., 2013. Linguistic Regularities in Continuous Space Word Representations. In NAACL-HLT.
- Mohamed, A., Yu, D., and Deng, L. 2010. Investigation of full-sequence training of deep belief networks for speech recognition, Proc. Interspeech.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. 2011. Multimodal deep learning, Proc. ICML.

Microsoft Research

References

- Sainath, T., Mohamed, A., Kingsbury, B., and Ramabhadran, B. 2013. Convolutional neural networks for LVCSR, Proc. ICASSP.
- Salakhutdinov R., and Hinton, G., 2007 Semantic hashing. in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models
- Sarikaya, R., Hinton, G., and Ramabhadran, B., 2011. Deep belief nets for natural language call-routing, in Proceedings of the ICASSP.
- Schwenk, H., Dchelotte, D., Gauvain, J-L., 2006. Continuous space language models for statistical machine translation, in COLING-ACL
- Seide, F., Li, G., and Yu, D. 2011. Conversational speech transcription using context-dependent deep neural networks, Proc. Interspeech
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. 2014. Learning Semantic Representations Using Convolutional Neural Networks for Web Search, in Proceedings of WWW.
- Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. 2014. A convolutional latent semantic model for web search. MSR Tech Report.
- Socher, R., Huval, B., Manning, C., Ng, A., 2012. Semantic compositionality through recursive matrix-vector spaces. In EMNLP.
- Socher, R., Lin, C., Ng, A., and Manning, C. 2011. Learning continuous phrase representations and syntactic parsing with recursive neural networks, Proc. ICML.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng A., and Potts. C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Proc. EMNLP
- Song, X. He, X., Gao. J., and Deng, L. 2014. Learning Word Embedding Using the DSSM. MSR Tech Report.
- Song, Y., Wang, H., and He, X., 2014. Adapting Deep RankNet for Personalized Search. Proc. WSDM.
- Tur, G., Deng, L., Hakkani-Tur, D., and He, X., 2012. Towards Deeper Understanding Deep Convex Networks for Semantic Utterance Classification, in ICASSP.
- Wright, S., Kanevsky, D., Deng, L., He, X., Heigold, G., and Li, H., 2013. Optimization Algorithms and Applications for Speech and Language Processing, in IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 11.
- Xu, P., and Sarikaya, R., 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling, in IEEE ASRU.
- Yao, K., Zweig, G., Hwang, M-Y. , Shi, Y., Yu, D., 2013. Recurrent neural networks for language understanding, submitted to Interspeech.
- Yann, D., Tur, G., Hakkani-Tur, D., Heck, L., 2014. Zero-Shot Learning and Clustering for Semantic Utterance Classification Using Deep Learning, in ICLR.
- Yih, W., Toutanova, K., Platt, J., and Meek, C. 2011. Learning discriminative projections for text similarity measures. In CoNLL.
- Zeiler, M. and Fergus, R. 2013. Visualizing and understanding convolutional networks, arXiv:1311.2901, pp. 1-11.