Deep Learning for NLP

Nancy McCracken

What is Deep Learning?

- Deep learning is a subfield of machine learning
  - Builds on Representation Learning to automatically learn good features/representations
  - Deep Learning algorithm learns multiple levels of feature representations in increasing levels of complexity or abstraction
- Deep learning can
  - Not only automatically learn good features
  - But do so by using vast amounts of unlabeled data

Overview material adapted from:
RS - Richard Socher, Stanford Course Notes, Deep Learning for NLP, 2016 and
MS - Manning and Socher, tutorial notes on Deep Learning, NAACL 2013.

Current Machine Learning

- Most current machine learning works well because humans design good input features
  - Example: features for finding named entities or organization names (Finkel, 2010)
- Machine learning solves an optimization problem over the feature space that learns weights for the features from labeled data, in order to make good predictions on new data
- Typical classification tasks required human annotated labeled data

Feature	NER
Current Word	✓
Previous Word	✓
Next Word	✓
Current Word Character n-gram	all
Current POS Tag	✓
Surrounding POS Tag Sequence	✓
Current Word Shape	✓
Surrounding Word Shape Sequence	✓
Presence of Word in Left Window	size 4
Presence of Word in Right Window	size 4

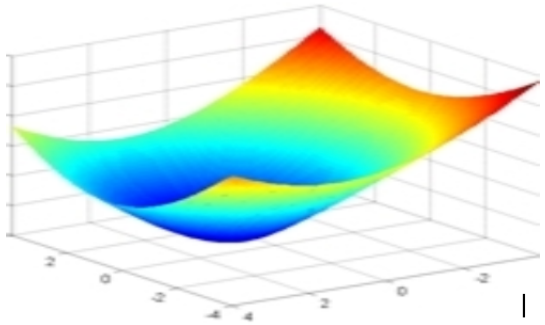


Diagram RS

Deep Learning Architecture

- Most commonly used architecture uses various types of multi-layer neural networks, such as Belief NN

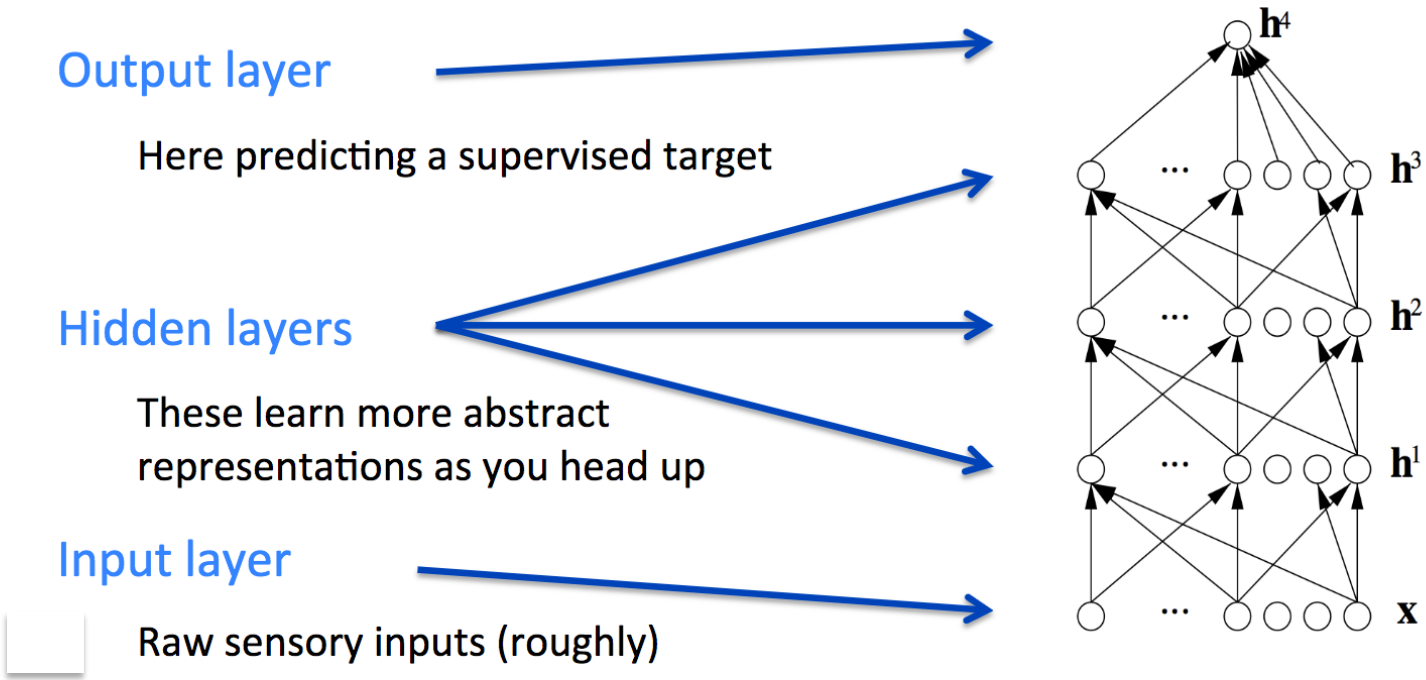


Diagram MS

Single Neuron

- A single neuron is a computational unit with an activation function (f). It takes inputs (3) plus a bias term (b) and gives an output
  - Expressed as a result h depending on weights w and b

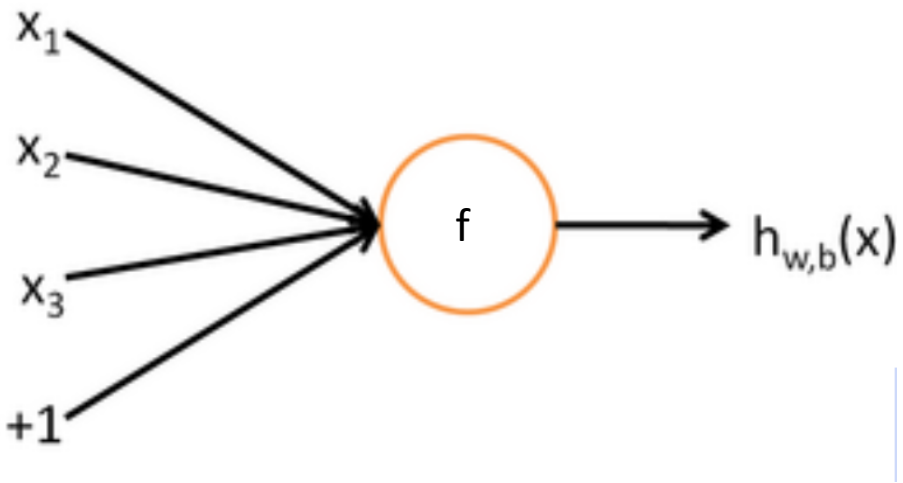


Diagram MS

Single Layer Neural Network (NN)

- Can also solve current machine learning problems
- If we feed a vector of inputs (the features) into a bunch of neurons then we get a vector of outputs, which can be combined according to the task we are trying to solve (the objective function h)

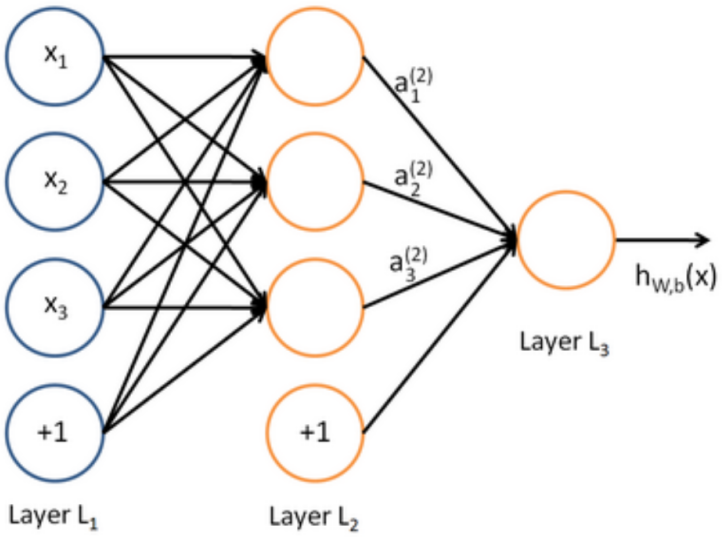


Diagram MS

Training the network learns the weights:

- run the network to predict an output,
- compare the output with the desired (gold) result,
- run back through the network adjusting the weights to reduce the error,
- and iterate.

The weights (and bias) gives the model to compute the predicted output for future data.

Deep Learning NN

- We can keep going and add multiple layers
- And we can revise our learning algorithm to also learn representations of the input X
  - Several algorithms for how to do the “feed forward” and “back propagation” in an efficient way

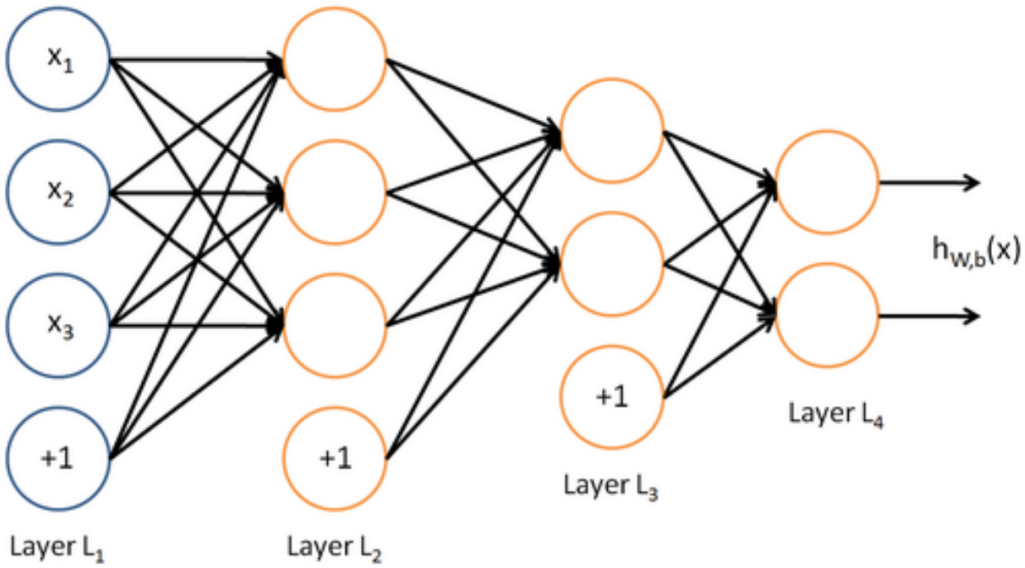


Diagram MS

Reasons for Deep Learning

- Break the bottleneck of manually designed features to automatically learn them
  - Easy to adapt and to use
- Can use large amounts of unsupervised data (e.g. raw text) to learn features and then use supervised data (with labels, like positive and negative) to learn a task
- Deep learning ideas have been known but only recently outperforming other techniques
  - Benefit from lots of data
  - Multi-core machines with faster processors
  - New models and algorithms

Deep Learning for Speech

- The first breakthrough results of deep learning on large datasets happened in speech recognition
- Context-dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition (Dahl et al 2010)

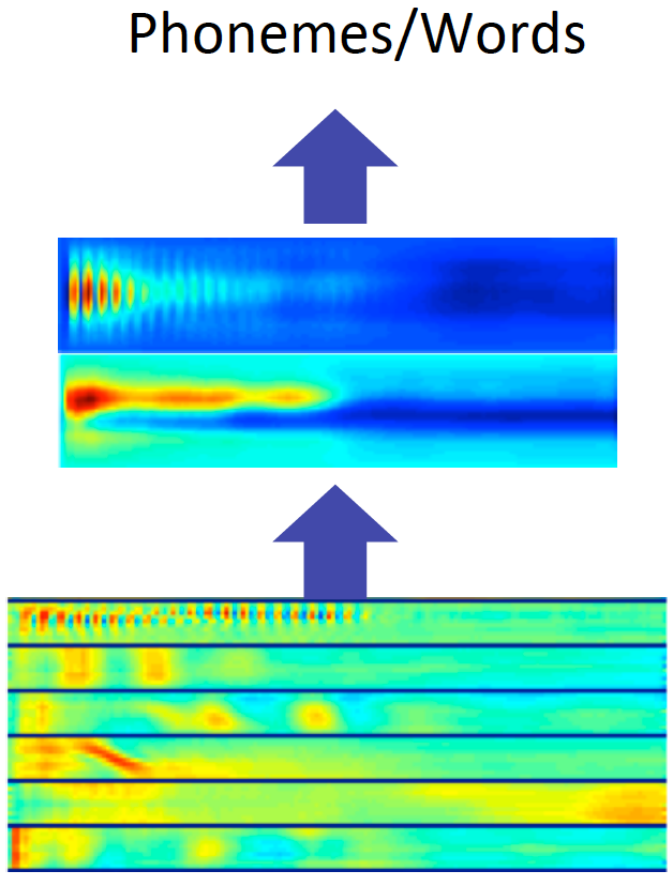


Diagram RS

DL Speech Results

- Compare state-of-the-art algorithm (GMM 40 mix BMMI) on 309 hours of Switchboard corpus with Deep Belief Network Deep NN with 7 layers by 2048
- Shows comparable reduction in error rates as the standard algorithm trained on 2000 hours of sound

MSR MAVIS Speech System

[Dahl et al. 2012; Seide et al. 2011; following Mohamed et al. 2011]

"The algorithms represent the first time a company has released a deep-neural-networks (DNN)-based speech-recognition algorithm in a commercial product."

Diagram MS

Acoustic model & training	Recog \ WER	RT03S FSH	Hub5 SWB
GMM 40-mix, BMMI, SWB 309h	1-pass −adapt	**27.4**	**23.6**
DBN-DNN 7 layer x 2048, SWB 309h	1-pass −adapt	**18.5** (−33%)	**16.1** (−32%)
GMM 72-mix, BMMI, FSH 2000h	*k*-pass +adapt	**18.6**	**17.1**

Results are error rates

Deep Learning for Vision

- Most of the earliest work in deep learning focused on computer vision
- Lee et all (2009) Zeiler and Fergus (2013)

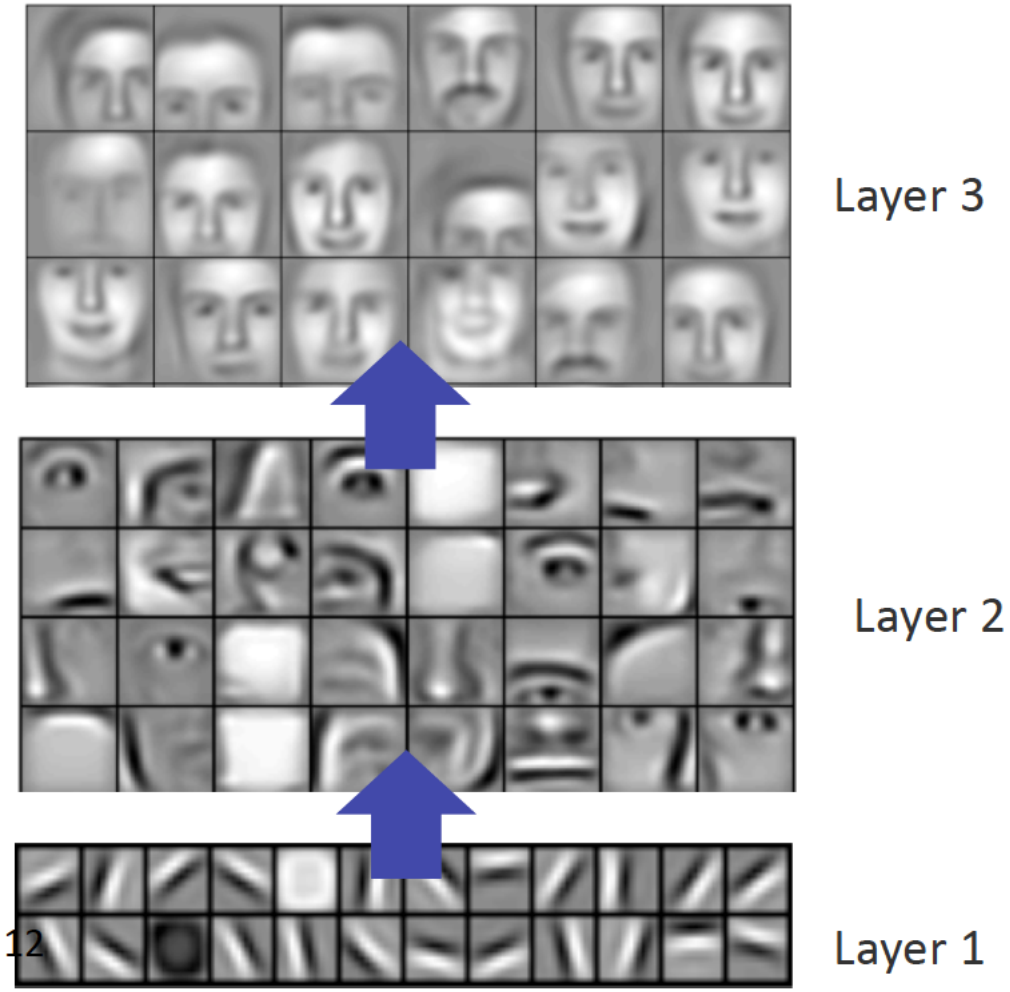
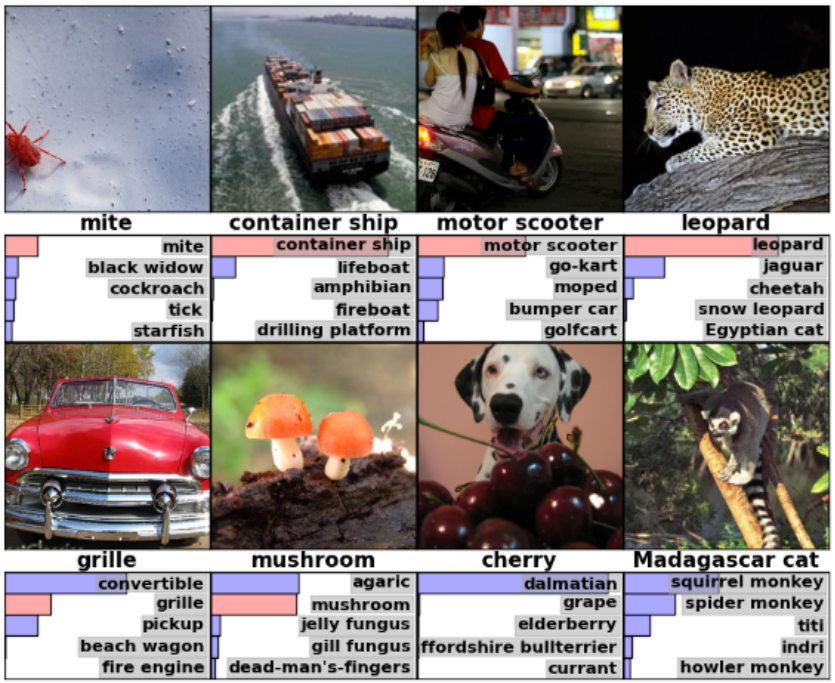


Diagram MS

DL Vision Results

- Breakthrough paper: ImageNet Classification with Deep Convolutional Neural Networks by Krizhevsky et al 2012



Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Results are error rates

NLP Word Representations

- Distributional similarity based representations
  - Representing a word by means of its neighbors
    - "You shall know a word by the company it keeps." (J.R. Firth 1957)
    - Or linguistic items with similar distributions have similar meanings
    - This idea is also in similarity measures such as Mutual Information
- One of the most successful ideas of modern statistical NLP

government debt problems turning into banking crises as has happened in

saying that Europe needs unified banking regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

You can vary whether you use local or large context to get a more syntactic or semantic clustering

Diagram MS

NN Dense Word Vectors

- Combine vector space semantics with probabilistic models to predict vectors of context words
  - (Bengio et al 2003, Collobert & Weston 2008, Turian et al 2010)
- A word is represented as a dense vector
- Older related ideas are
  - SVD on term-context matrix
  - Brown clusters

linguistics =

$$\begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Diagram MS

NN Learning Dense Word Vectors

- Set up a classification task from unsupervised data where we have positive training examples directly from the data, and negative examples obtained by substituting a random word in the context (as described in Collobert et al JMLR 2011)
  - Positive example: "cat sits on the *mat*"
  - Negative example: "cat sits jeju the *mat*"
- Classify which contexts are noise

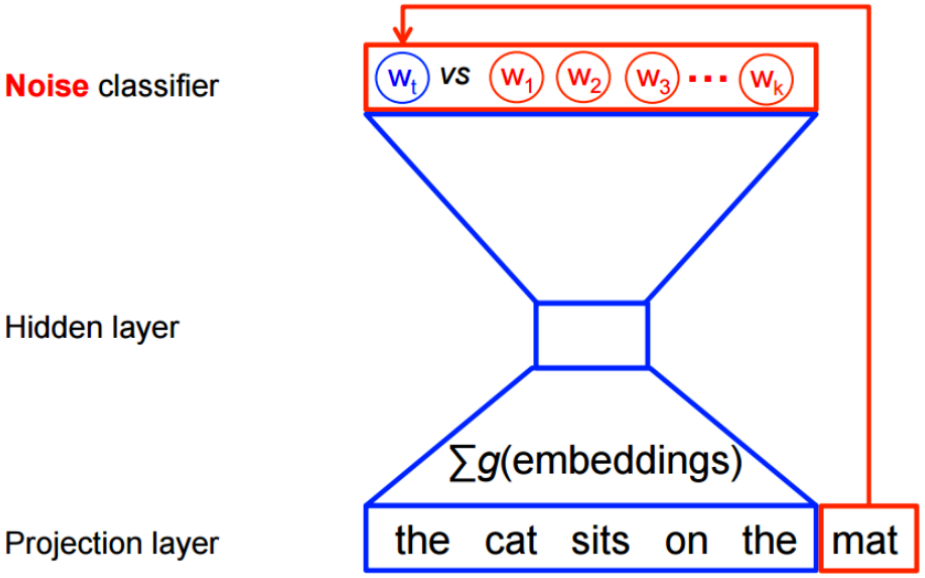


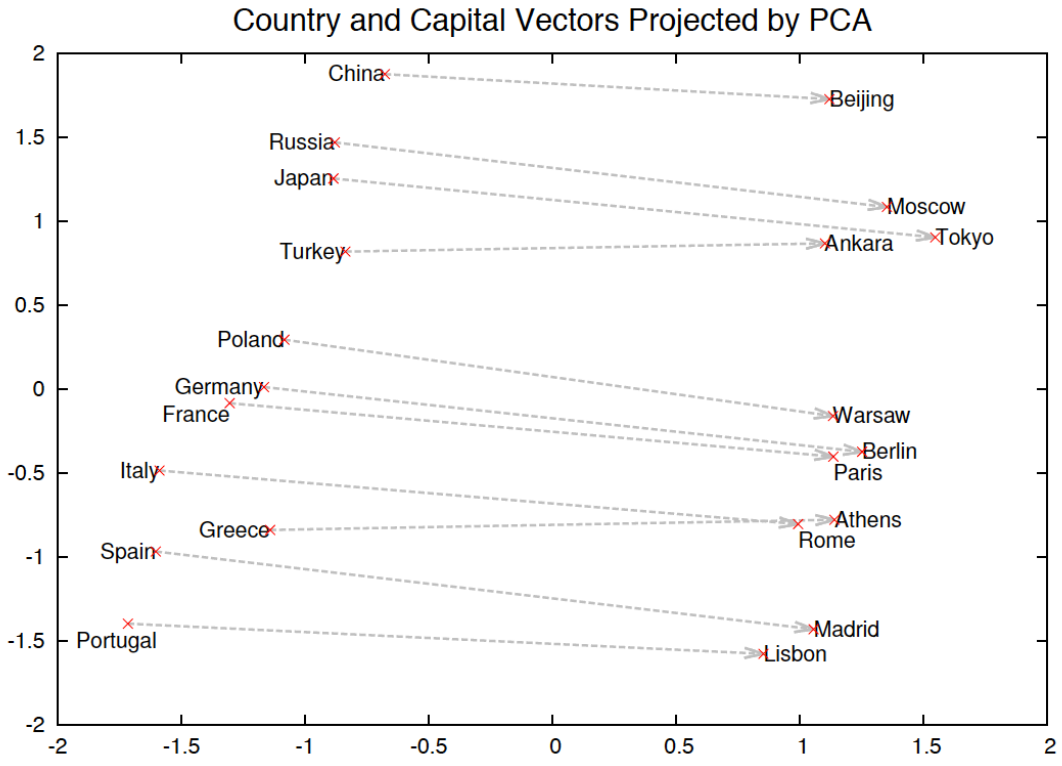
Diagram TensorFlow tutorial

Word2Vec

- The word vector classifier gives a simpler and faster implementation of a (shallow) RNN, (Mikolov 2013) with 2 algorithms
  - CBOW (continuous bag of words) predicts the current word w, given the neighboring words in the window
  - SkipGram predicts the neighboring words, given w
- Allows the NN to be applied to large amounts of data
- Hyperparameters
  - Window size – the number of context words
  - Network size – the number of neurons in the hidden layer
  - Other parameters such as negative subsampling number

Dense Word Vector Space

- In the resulting space, similar words should be closer together
  - Syntactic similarities, such as word tense or plurals
  - Semantic similarities



Length 1000 vectors projected to 2D, diagram from Mikolov et al 2013 (NIPS)

Dense Word Vector Space

- Showing some of the nearest words in the vector space (Mikolov 2013)

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

Analogies Task

- How can we evaluate whether the dense word vectors represent good word similarities?
- Solve problems of the type:
  - "*a* is to *b* as *c* is to __"
- Mikolov et al (HLT 2013) constructed a test set of 8k syntactic relations
  - Noun plurals and possessives, verb tenses, adjectival comparitives and superlatives
- Semantic test set from Semeval-2012 Task 2

Word Relationships

- Mikolov's results are that analogies testing dimensions of similarity can do quite well just by doing vector subtractions
  - Syntactically – plurals, verb tenses, adjective forms

$$x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families}$$

  - Semantically (analogies from Semeval 2012 task 2)

$$x_{shirt} - x_{clothing} \approx x_{chair} - x_{furniture}$$

$$x_{king} - x_{man} \approx x_{queen} - x_{woman}$$

Diagram RS

Word Analogies

- Results from Mikolov et al 2013 (HLT) using word2vec
  - Trained on 320M words of broadcast news data
  - With 82k word vocabularv

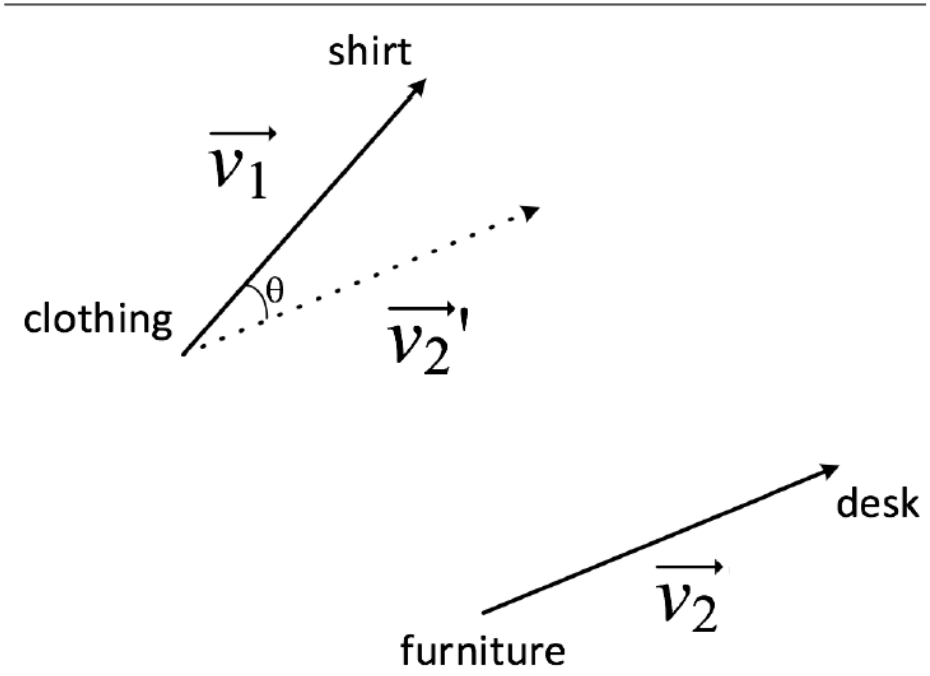


Diagram MS

Method	Syntax % correct
LSA 320 dim	16.5 [best]
RNN 80 dim	16.2
RNN 320 dim	28.5
RNN 1600 dim	39.6
Method	**Semantics Spearm ρ**
UTD-NB (Rink & H. 2012)	0.230 [Semeval win]
LSA 640	0.149
RNN 80	0.211
RNN 1600	0.275 [new SOTA]

NLP Word Level Classifiers

- Similar to word vector learning, but replaces single vector output with a classifier layer for the task, S
- Diagram is similar to a conventional classifier, except that it includes learning for the feature input vectors from unsupervised data

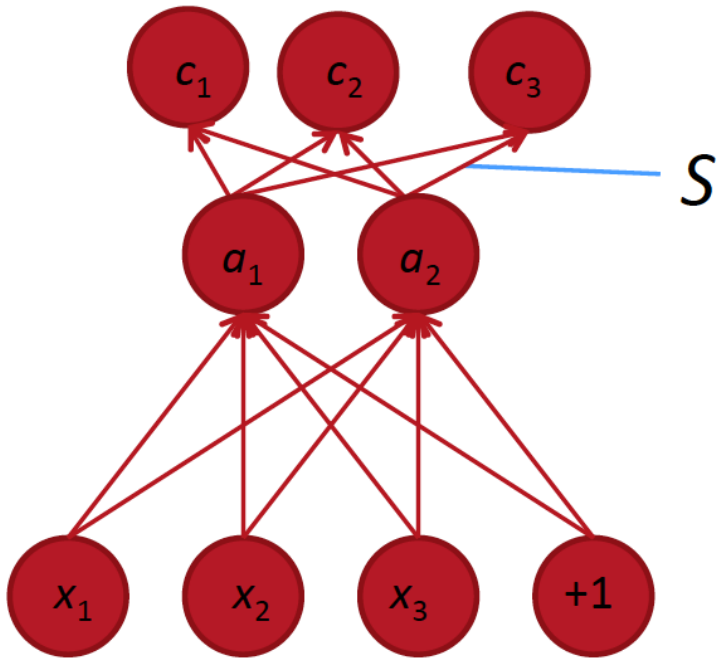


Diagram MS

Two NLP word level tasks

- POS tagging and Named Entity Recognition (NER)

	POS WSJ (acc.)	NER CoNLL (F1)
State-of-the-art*	97.24	89.31
Supervised NN	**96.37**	**81.47**
Unsupervised pre-training followed by supervised NN**	**97.20**	**88.87**
+ hand-crafted features***	97.29	89.59

* Representative systems: POS: (Toutanova et al. 2003), NER: (Ando & Zhang 2005)

** 130,000-word embedding trained on Wikipedia and Reuters with 11 word window, 100 unit hidden layer – **for 7 weeks!** – then supervised task training

***Features are character suffixes for POS and a gazetteer for NER

Diagram MS

Architectures for NLP tasks with Structures

- Would like a Deep Learning approach that can use good intermediate representations that can be shared across tasks
- Many NLP language tasks share syntactic sentence structure
- These structures are also recursive in nature => Recursive Deep Learning

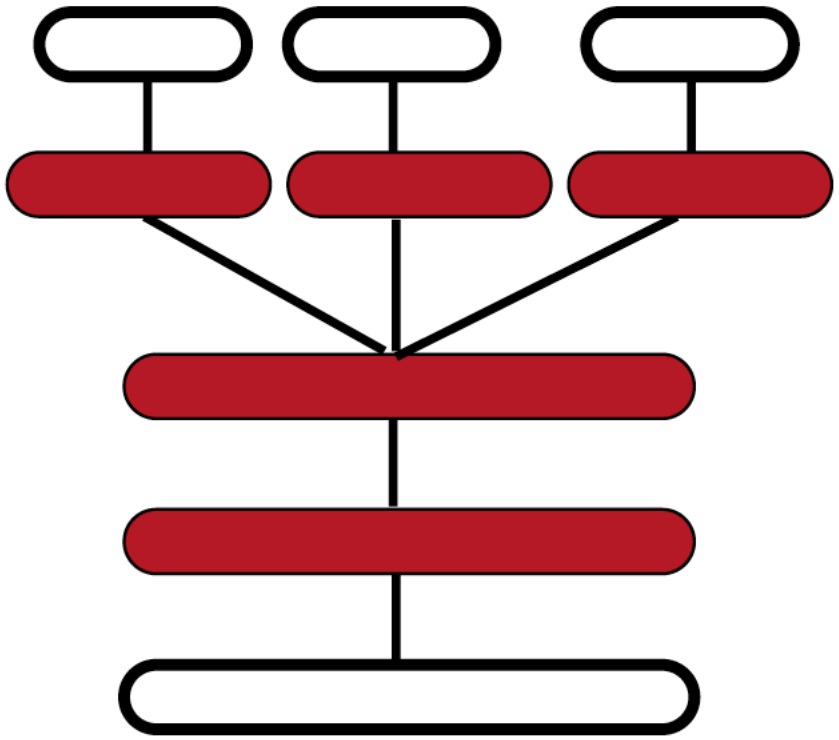
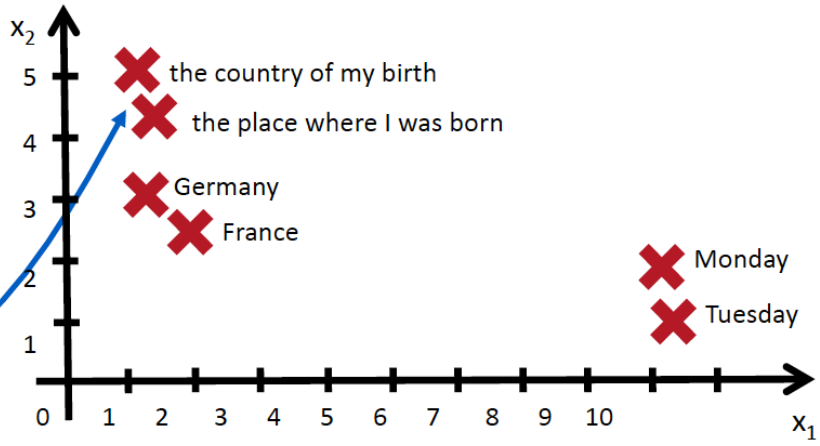


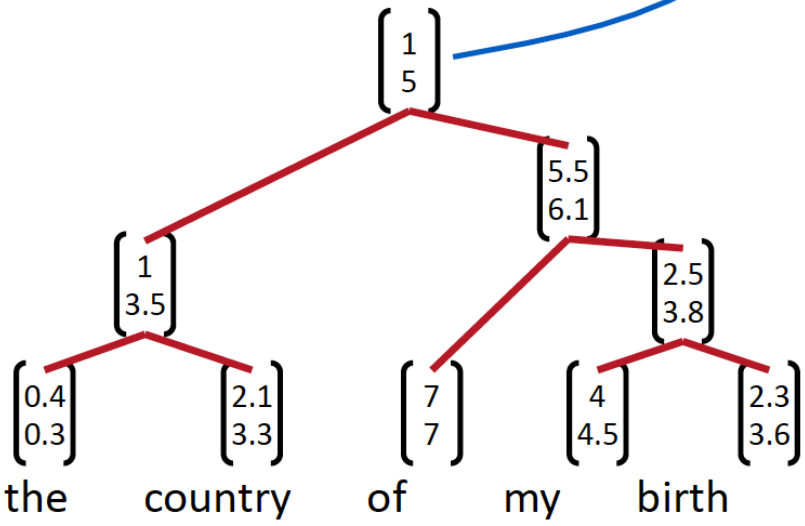
Diagram MS – one design of NN layer representations applied to several tasks

Phrase Level Vectors

- Represent the meaning of longer phrases by mapping them into the same vector space

Use principle of compositionality
The meaning (vector) of a sentence is determined by (1) the meanings of its words and (2) the rules that combine them.





Models in this section can jointly learn parse trees and compositional vector representations

Diagram RS

Recursive Neural Networks

- Instead of traditional combination of NN node output by summing weighted vectors, combine two semantic representations and score how plausible the result will be

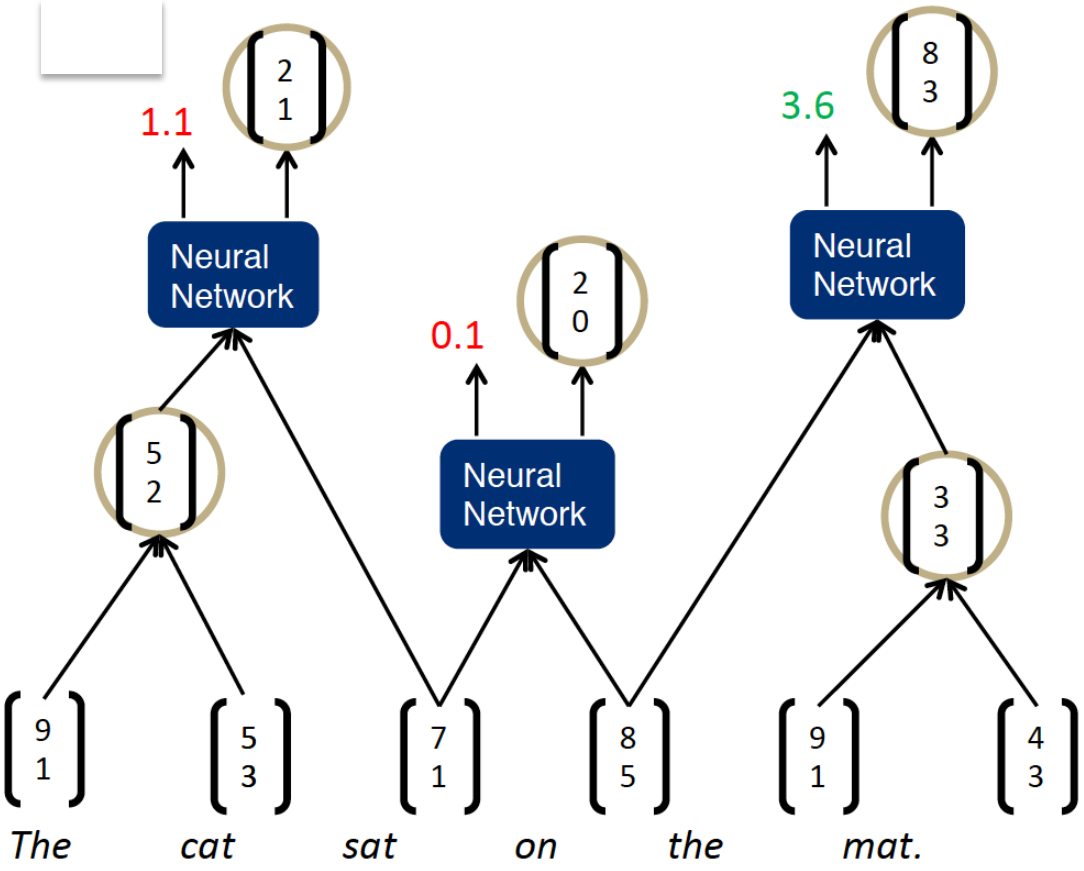


Diagram MS – In parsing "the cat sat on the mat", combining vectors representing phrases to get new phrase vector and score

Parsing Results

- Tested on standard WSJ with F1 scores
- CVG is RNN combined with PCFG (probabilistic context free grammars) , SU is syntactically untied RNN (faster)

Parser	Test, All Sentences
Stanford PCFG, (Klein and Manning, 2003a)	85.5
Stanford Factored (Klein and Manning, 2003b)	86.6
Factored PCFGs (Hall and Klein, 2012)	89.4
Collins (Collins, 1997)	87.7
SSN (Henderson, 2004)	89.4
Berkeley Parser (Petrov and Klein, 2007)	90.1
CVG (RNN) (Socher et al., ACL 2013)	85.0
CVG (SU-RNN) (Socher et al., ACL 2013)	90.4
Charniak - Self Trained (McClosky et al. 2006)	91.0
Charniak - Self Trained-ReRanked (McClosky et al. 2006)	92.1

Advantage of Deep Learning

- Current NLP systems, in general, are fragile because they depend on which lexical items are in the supervised training data
- Add robustness with word representations from unlabeled data
  - WSJ doesn't have many occurrences of foods, almost none of "bananas"
  - But word vector for "bananas" is similar to "oranges", get a better parse

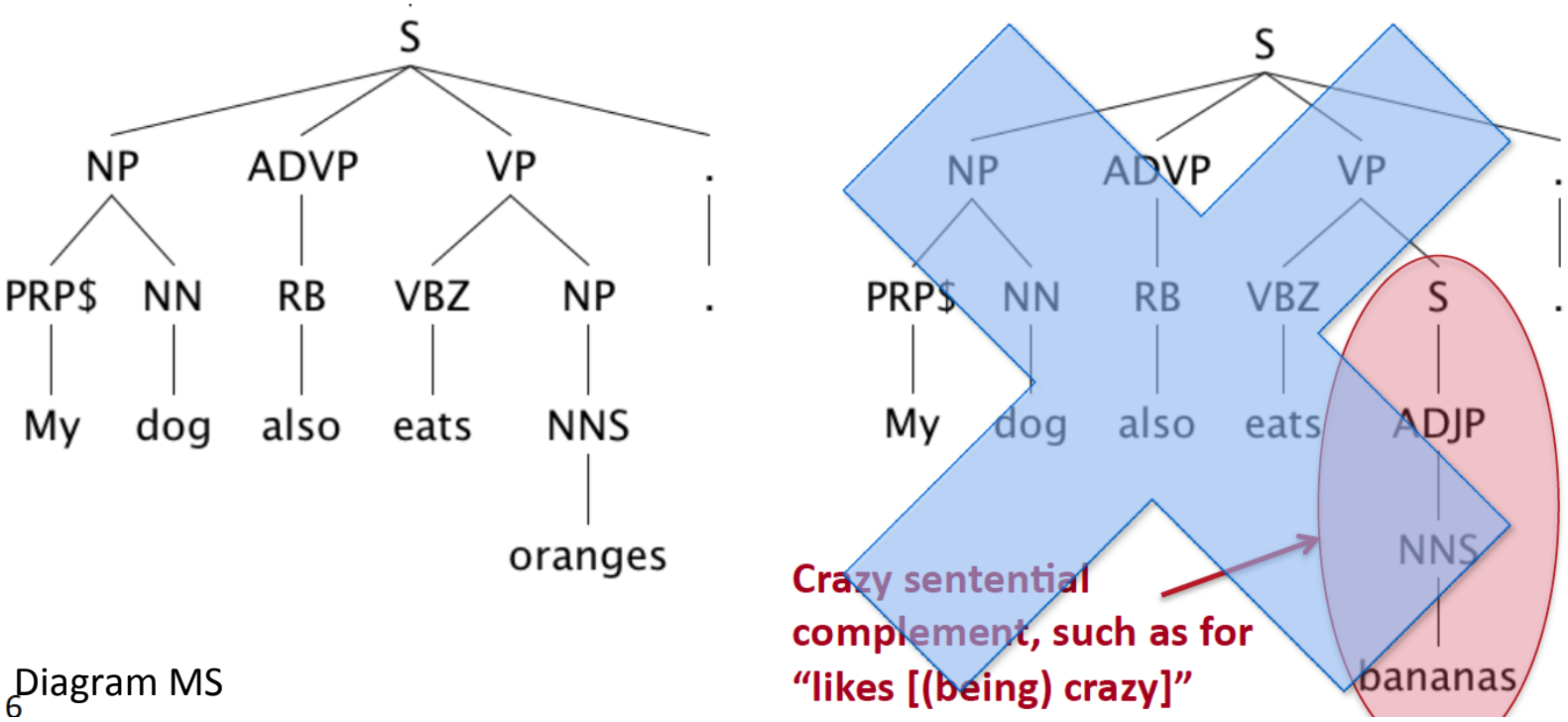


Diagram MS

Paraphrase Detection

- Task is to compare sentences to see if they have the same semantics
- Examples
  - Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses
  - Basically, the plaintiffs did not show that omissions in Merrill's research caused the claimed losses
  - The initial report was made to Modesto Police in Decembr 28
  - It stems from a Modesto police report
- Solution is a RNN called Recursive Autoencoders to compare vector representations of sentences

Paraphrase Results

- Experiments on Microsoft Paraphrase Corpus

Method	Acc.	F1
Rus et al.(2008)	70.6	80.5
Mihalcea et al.(2006)	70.3	81.3
Islam et al.(2007)	72.6	81.3
Qiu et al.(2006)	72.0	81.6
Fernando et al.(2008)	74.1	82.4
Wan et al.(2006)	75.6	83.0
Das and Smith (2009)	73.9	82.3
Das and Smith (2009) + 18 Surface Features	76.1	82.7
F. Bu et al. (ACL 2012): String Re-writing Kernel	76.3	--
Unfolding Recursive Autoencoder (NIPS 2011)	**76.8**	**83.6**

Sentiment Analysis on Movies

- Label movie review sentences for positive or negative sentiment
- Examples
  - Stealing Harvard doesn't care about cleverness, wit, or any other kind of intelligent humor.
  - There are slow and repetitive parts, but it has just enough spice to keep it going.
- Solution:
  - New sentiment phrase treebank to provide supervised means of combining sentiment phrases (available from Socher and Kaggle)
  - Recursive Neural Tensor Network – most powerful method so far and provides more interaction of vectors in the composition of phrases (Socher et al 2013)

Sentiment Results

- Evaluate results on sentences from movie reviews (Pang and Lee 2006)

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.6**	**87.6**	**85.4**

Diagram RS

Visualization of Sentiment

- RNTN for sentiment can capture “X but Y” (shown here) and also various negation constructs

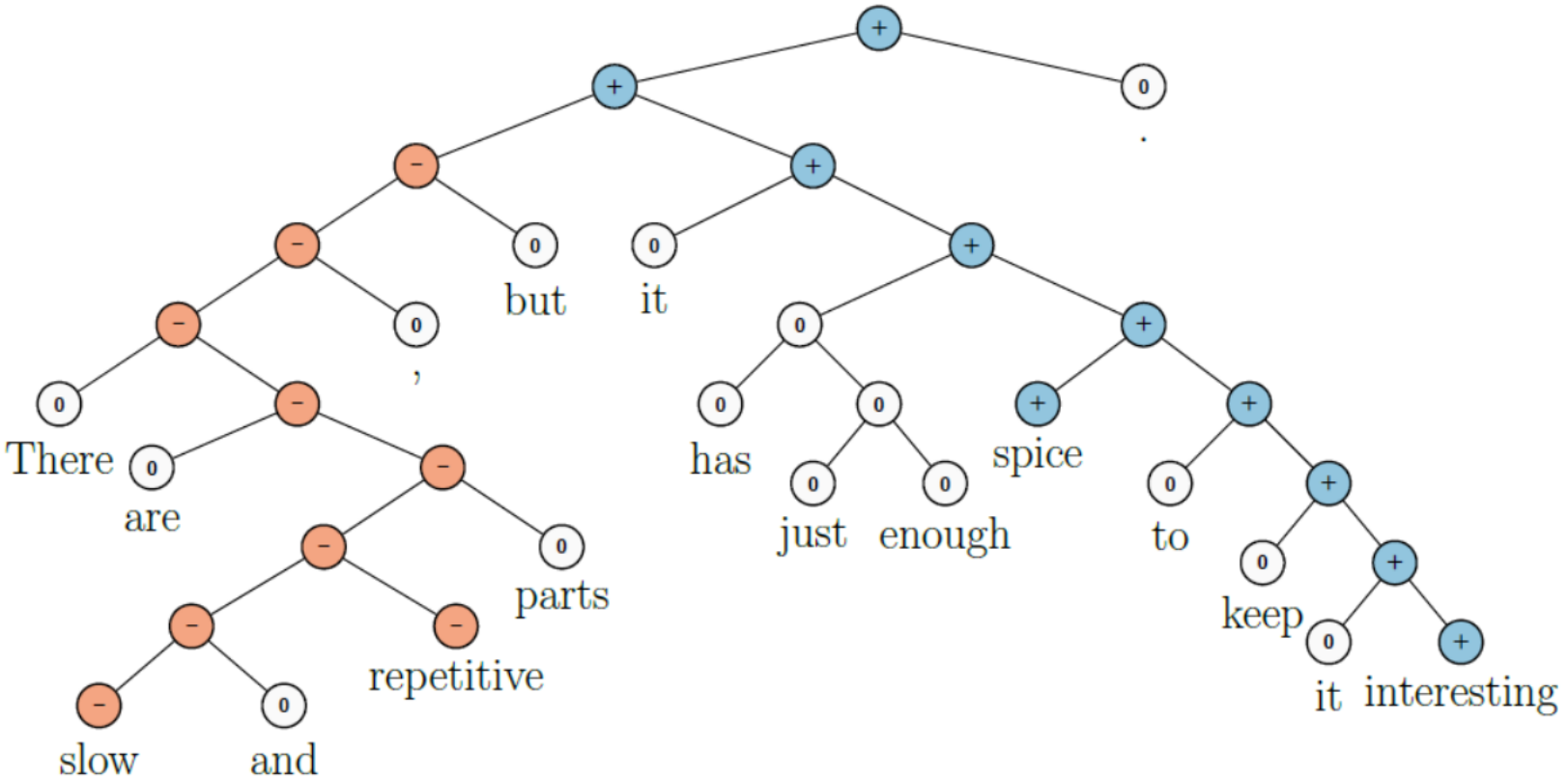


Diagram RS

Summary of Deep Learning on NLP tasks

- Also RNN for
  - Relationship learning
  - Question answering
  - Object detection in images
- Excellent recent results on Machine Translation has now gone into products such as Google Translate
  - Sequence to Sequence Learning with NN
  - Sutskever et al 2014, Luong et al 2016
- Ongoing research into types of NN and how to apply them to tasks

How can we use Deep Learning in NLP?

- (In the ACL 2013 tutorial, Manning and Socher gave advice on how to write your own neural network code!)
- Code available for sentiment and relation analysis: www.socher.org
- Tools for word representations
  - word2vec, available from Google in C
  - gensim (python), open source by Radim Rehurek
  - Deeplearning4j (java)
- Use word vectors as features in current classifiers
  - doc2vec
- Google trained word model – trained on 100B words news data resulting in 3M phrases with layer size 300
- More advanced types of NN
  - TensorFlow, available from Google

gensim package Word2vec

- From RaRe technologies, Radim Rehurek
  - https://rare-technologies.com/word2vec-tutorial/
- Create a generator to get sentences as lists of tokens
  - Do whatever preprocessing and tokenizing you need
- Create a model with Word2vec
  - Set minimum word frequency for vocabulary
  - Set window size (surrounding context word size)
  - Set layer size (size of hidden layer and length of vectors)
  - Set number of cores for parallel processing
  - Recommended to try window sizes from 5 to 20, and layer sizes from 50 to 300
  - Recommend CBOW for smaller datasets, Skipgram for larger
- Evaluate analogies with Google's file: questions-words.txt
- Other functions show similar, and dis-similar, words

Word2vec on compaign social media (in progress!)

- I created a word2vec model on 114k combined twitter and facebook messages (unlabeled)
  - Used NLTK social media tokenizer; didn't clean urls, hashtags or mentions
  - Model has window size 8 and layer size 200
- Vocabulary is size 30,342 with minimum count = 3
  - Examples: ['benefits', 'twenty', 'according', 'will', 'Steps', '@obrienc2', '@HoustonChron', 'Balloon', '#actionsnotwords', 'owe', 'http://t.co/adp2vJr5u7', 'Texan', 'possibility', 'initiating', '!', 'humble', '@anniekarni', '7:40', 'chip-in', 'shouting', '@masslivenews', 'Timothy', 'strategically', 'racist', '10/8', 'reigniting', 'Jimmie', 'Too', '@Teamsters',
- Similarities:
  - modelg.most_similar(['pollster']) : 'groups', 'attended', '@NancyPelosi'
  - modelg.doesnt_match(["candidate" "pollster" "wives" "paper"): 'paper'
  - modelg.similarity('candidate', 'pollster')) : 0.06
  - modelg.similarity('paper', 'pollster') : - 0.08

Paragraph level vectors

- In order to use word dense vectors as features in a current classifier, use RNN to combine word vectors for entire paragraph, sentence or (short) document
- Le and Mokolov (JMLR 2014)
  - Training to get word vectors
  - Training to get paragraph vectors, combining word vectors w/ paragraph, learning the appropriate weights
  - Inference function to get vectors on new paragraphs – fixes the learned weights and combines the doc with word vectors to get vector rep of doc

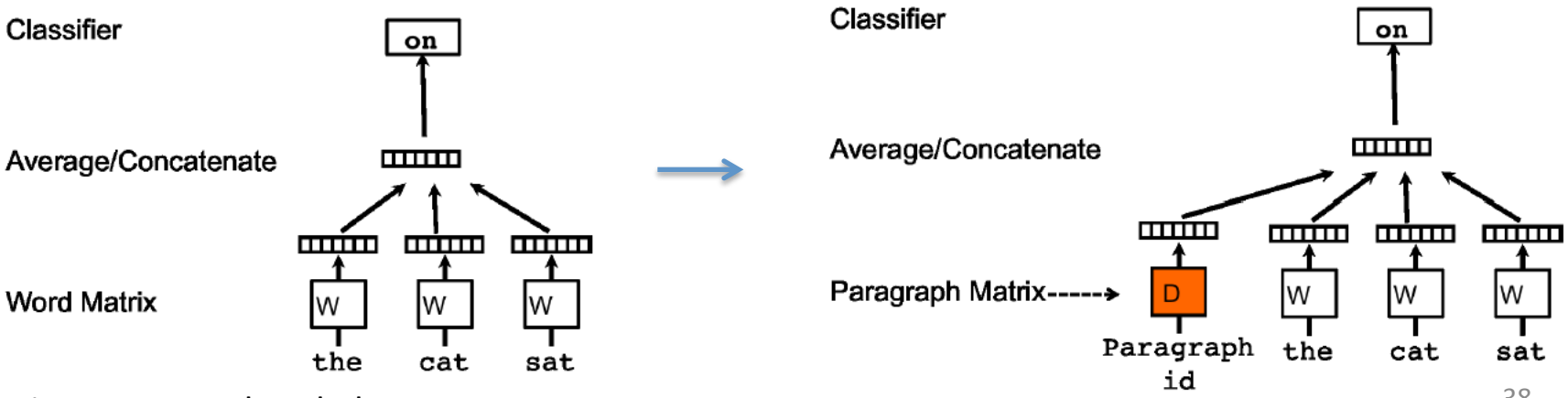


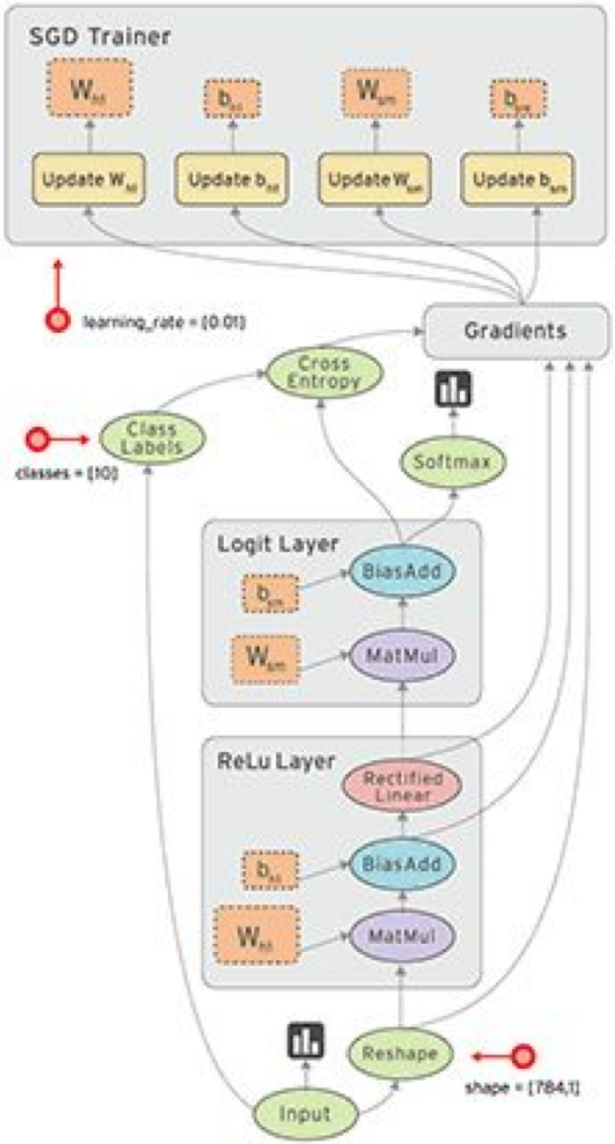
Diagram Le and Mokolov

gensim package Doc2vec

- From RaRe technologies, Radim Rehurek
  - https://radimrehurek.com/gensim/models/doc2vec.html
- Create a generator that pairs ids with sentences
- Create a model with Doc2vec
  - Similar parameters as Word2vec
  - Save the model
- In your classifier setup, load the model
  - Call the infer_vector function on the token list to get a (dense) vector representation
  - Add these numbers to your features and classify

TensorFlow

- Framework from Google for scalable Machine Learning
- Define computation as a graph
  - Graph is defined with python fns
  - Compiled, optimized, executed
  - Nodes represent computations
  - Data (tensors) flow along edges
- Google refactored their ML platform
  - Framework of reusable components
- Manage distributed, heterogeneous systems



TensorFlow slides from Martin Wicke, ACM webinar 2016

What's available in TensorFlow now

- Tutorials on tensorflow.org:
  - Image recognition (convolutional NN):
    - https://www.tensorflow.org/tutorials/image_recognition
  - Word embeddings:
    - https://www.tensorflow.org/versions/word2vec
  - Language Modeling:
    - https://www.tensorflow.org/tutorials/recurrent
  - Translation:
    - https://www.tensorflow.org/versions/seq2seq
- Lynda tutorial on deep learning
  - All SU ids can access a Lynda account

Conclusion

- Deep learning is promising area for NLP
  - Learning features of classification from unlabeled data
  - But may not totally free us from designing manual features and using labeled data to make NLP representations
- Lots of research on what are the right NN algorithms and text representations
- Future:
  - More software packages to make it easier to apply DL to your own NLP task
  - Possibilities for improvements in some of the harder tasks of NLP

References

- Ando, Rie Kubota and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. J. Machine Learning Research 6:1817–1853.
- Bengio, Yoshua, Ŕejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. J. Machine Learning Research 3:1137–1155.
- Collobert, R. and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In ICML'2008.
- Collobert, Ronan, Jason Weston, Ĺeon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research 12:2493–2537.
- Dahl, George E., Dong Yu, Li Deng, and Alex Acero. 2012. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing 20(1):33–42.
- Finkel, Jenny Rose and Christopher D. Manning. 2010. Hierarchical Joint Learning: Improving Joint Parsing and Named Entity Recognition with Non-Jointly Labeled Data, HLT

References

- Firth, J. R. 1957. A synopsis of linguistic theory 1930–1955. In Studies in Lin-guistic Analysis, pages 1–32. Oxford: Philological Society. Reprinted in F.R. Palmer (ed), Selected Papers of J. R. Firth 1952–1959, London: Longman, 1968.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E., ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada
- Le, QV, T Mikolov, Distributed Representations of Sentences and Documents, ICML 14, 1188-1196
- Luong, Minh-Thang, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Lukasz Kaiser. Multi-task Sequence to Sequence Learning. *ICLR'16*.
- Manning, Christopher and Socher, Richard, Deep Learning for NLP without Magic, Tutorial for HLT NAACL 2013.
- Mikolov, Tomas. W Yih, G Zweig, Linguistic Regularities in Continuous Space Word Representations, HLT-NAACL 13, 746-751
- Mikolov, T., I Sutskever, K Chen, GS Corrado, J Dean, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, 3111-3119

References

- Seide, Frank, Gang Li, and Dong Yu. 2011. Conversational speech transcription using context-dependent deep neural networks. In Interspeech 2011, pages 437–440
- Socher, R., A Perelygin, JY Wu, J Chuang, CD Manning, AY Ng, CP Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, EMNLP 2013.
- Socher, Richard, Deep Learning for NLP class , http://cs224d.stanford.edu/
- Sutskever,Ilya, Oriol Vinyals, and Quoc Le, Sequence to Sequence Learning with Neural Networks, NIPS 2014
- Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003), pages 252–259.
- Turian, Joseph, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In Proc. ACL'2010, pages 384–394. Association for Computational Linguistics.
- Zeiler, Matthew D. and Rob Fergus, Visualizing and Understanding Convolutional Networds, ECCV 2014.