

NLP Applications

Deep Learning Lecture 8

Samuel Cheng

School of ECE
University of Oklahoma

Spring, 2017
(Slides credit to Stanford CS224d)

Table of Contents

- 1 RNN word based language model
- 2 Name Entity Recognition
- 3 Recursive neural networks
- 4 Conclusions

- Activity 1 due on Sunday
 - 5% penalty per day starting next Monday
- Siraj is the winner for Activity 1
- Ahmad and Tamer will present today. Please vote accordingly
 - Some hiccups with the voting system last week. Should be fixed now
 - To address Naim's concern, you will now need to input your passcode to vote. You should have received your passcode through email yesterday (from phsamuel2016@gmail.com)
- Thanks to Muhanad, Caffe is up on Schooner
- Maybe quick demo of pbworks during break

- Activity 1 due on Sunday
 - 5% penalty per day starting next Monday
- Siraj is the winner for Activity 1
- Ahmad and Tamer will present today. Please vote accordingly
 - Some hiccups with the voting system last week. Should be fixed now
 - To address Naim's concern, you will now need to input your passcode to vote. You should have received your passcode through email yesterday (from phsamuel2016@gmail.com)
- Thanks to Muhanad, Caffe is up on Schooner
- Maybe quick demo of pbworks during break

- Activity 1 due on Sunday
 - 5% penalty per day starting next Monday
- Siraj is the winner for Activity 1
- Ahmad and Tamer will present today. Please vote accordingly
 - Some hiccups with the voting system last week. Should be fixed now
 - To address Naim's concern, you will now need to input your passcode to vote. You should have received your passcode through email yesterday (from phsamuel2016@gmail.com)
- Thanks to Muhanad, Caffe is up on Schooner
- Maybe quick demo of pbworks during break

- Activity 1 due on Sunday
 - 5% penalty per day starting next Monday
- Siraj is the winner for Activity 1
- Ahmad and Tamer will present today. Please vote accordingly
 - Some hiccups with the voting system last week. Should be fixed now
 - To address Naim's concern, you will now need to input your passcode to vote. You should have received your passcode through email yesterday (from phsamuel2016@gmail.com)
- Thanks to Muhanad, Caffe is up on Schooner
- Maybe quick demo of pbworks during break

Project proposal

- Submit your team info, project title, and abstract by the class of the week after spring break (3/23)
 - 5% of the total course
- Group projects are graded the same as single person projects. Given more hands there, a slight penalty is imposed for small group but goes steep as size increases (out of 40)

# members in group	2	3	4	5
Penalty	-2	-4	-8	-16

- Additional bonus (4% overall) if the projects lead to a submitted publication before course ends

Class rescheduling reminder

- Next two classes after spring break will be on 3/23 and 3/30 (Thursday for both) and the time will start at 4:30 pm instead of 3:30 pm
- Will send you all another reminder for actual location

Review and Overview

- We looked into echo-state networks and also image captioning last time
- More discussion on NLP applications this time

RNN word based language model

Tom Mikolov et al. 2010

- We talked about character based language model two classes ago. Let's look into a word based language model today
- Just as the character based model, we can use recurrent unit to memorize the information up to the current word/character
 - However, the word alphabet size can be huge
- This alphabet size problem can be solved using word embedding (word2vec)
 - For any word, first code the word using one-hot encoding
 - That is, a "delta" vector with length of the total number of words. And every component is zero except the location corresponding to the input word
 - Multiply the one-hot vector by an embedding matrix L . The embedding matrix shrink the dim of the one-hot vector

RNN word based language model

Tom Mikolov et al. 2010

- We talked about character based language model two classes ago. Let's look into a word based language model today
- Just as the character based model, we can use recurrent unit to memorize the information up to the current word/character
 - However, the word alphabet size can be huge
- This alphabet size problem can be solved using word embedding (word2vec)
 - For any word, first code the word using one-hot encoding
 - That is, a "delta" vector with length of the total number of words. And every component is zero except the location corresponding to the input word
 - Multiply the one-hot vector by an embedding matrix L . The embedding matrix shrink the dim of the one-hot vector

RNN word based language model

Tom Mikolov et al. 2010

- We talked about character based language model two classes ago. Let's look into a word based language model today
- Just as the character based model, we can use recurrent unit to memorize the information up to the current word/character
 - However, the word alphabet size can be huge
- This alphabet size problem can be solved using word embedding (word2vec)
 - For any word, first code the word using one-hot encoding
 - That is, a "delta" vector with length of the total number of words. And every component is zero except the location corresponding to the input word
 - Multiply the one-hot vector by an embedding matrix L . The embedding matrix shrink the dim of the one-hot vector

RNN word based model

Tom Mikolov et al. 2010

- x_t is word at t position and $\mathbf{x}^{(t)}$ is its one-hot **row**-vector representation
- Vocab size: $|V|$; Embedding dim: d ; Hidden Layer dim: D_h

$$\mathbf{e}^{(t)} = \mathbf{x}^{(t)} L$$

$$\mathbf{h}^{(t)} = \text{sigmoid}(\mathbf{h}^{(t-1)} H + \mathbf{e}^{(t)} I + \mathbf{b}_1)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{h}^{(t)} U + \mathbf{b}_2)$$

$$p(x_{t+1} = v_j | x_t, \dots, x_1) = \hat{y}_j^{(t)}$$

Note that $L \in \mathbb{R}^{|V| \times d}$, $H \in \mathbb{R}^{D_h \times D_h}$, $I \in \mathbb{R}^{d \times D_h}$, $U \in \mathbb{R}^{D_h \times |V|}$, $\mathbf{b}_1 \in \mathbb{R}^{D_h}$, $\mathbf{b}_2 \in \mathbb{R}^{|V|}$, where $d \ll |V|$

RNN word based model

Tom Mikolov et al. 2010

- x_t is word at t position and $\mathbf{x}^{(t)}$ is its one-hot **row**-vector representation
- Vocab size: $|V|$; Embedding dim: d ; Hidden Layer dim: D_h

$$\mathbf{e}^{(t)} = \mathbf{x}^{(t)} L$$

$$\mathbf{h}^{(t)} = \text{sigmoid}(\mathbf{h}^{(t-1)} H + \mathbf{e}^{(t)} I + \mathbf{b}_1)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{h}^{(t)} U + \mathbf{b}_2)$$

$$p(x_{t+1} = v_j | x_t, \dots, x_1) = \hat{y}_j^{(t)}$$

Note that $L \in \mathbb{R}^{|V| \times d}$, $H \in \mathbb{R}^{D_h \times D_h}$, $I \in \mathbb{R}^{d \times D_h}$, $U \in \mathbb{R}^{D_h \times |V|}$, $\mathbf{b}_1 \in \mathbb{R}^{D_h}$, $\mathbf{b}_2 \in \mathbb{R}^{|V|}$, where $d \ll |V|$

RNN word based model

Tom Mikolov et al. 2010

- x_t is word at t position and $\mathbf{x}^{(t)}$ is its one-hot **row**-vector representation
- Vocab size: $|V|$; Embedding dim: d ; Hidden Layer dim: D_h

$$\mathbf{e}^{(t)} = \mathbf{x}^{(t)} L$$

$$\mathbf{h}^{(t)} = \text{sigmoid}(\mathbf{h}^{(t-1)} H + \mathbf{e}^{(t)} I + \mathbf{b}_1)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{h}^{(t)} U + \mathbf{b}_2)$$

$$p(x_{t+1} = v_j | x_t, \dots, x_1) = \hat{y}_j^{(t)}$$

Note that $L \in \mathbb{R}^{|V| \times d}$, $H \in \mathbb{R}^{D_h \times D_h}$, $I \in \mathbb{R}^{d \times D_h}$, $U \in \mathbb{R}^{D_h \times |V|}$, $\mathbf{b}_1 \in \mathbb{R}^{D_h}$, $\mathbf{b}_2 \in \mathbb{R}^{|V|}$, where $d \ll |V|$

RNN word based model

Tom Mikolov et al. 2010

- x_t is word at t position and $\mathbf{x}^{(t)}$ is its one-hot **row**-vector representation
- Vocab size: $|V|$; Embedding dim: d ; Hidden Layer dim: D_h

$$\mathbf{e}^{(t)} = \mathbf{x}^{(t)} L$$

$$\mathbf{h}^{(t)} = \text{sigmoid}(\mathbf{h}^{(t-1)} H + \mathbf{e}^{(t)} I + \mathbf{b}_1)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{h}^{(t)} U + \mathbf{b}_2)$$

$$p(x_{t+1} = v_j | x_t, \dots, x_1) = \hat{y}_j^{(t)}$$

Note that $L \in \mathbb{R}^{|V| \times d}$, $H \in \mathbb{R}^{D_h \times D_h}$, $I \in \mathbb{R}^{d \times D_h}$, $U \in \mathbb{R}^{D_h \times |V|}$, $\mathbf{b}_1 \in \mathbb{R}^{D_h}$, $\mathbf{b}_2 \in \mathbb{R}^{|V|}$, where $d \ll |V|$

RNN word based model

Tom Mikolov et al. 2010

- We will try to minimize cost entropy as usual. Namely,

$$CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = - \sum_{i=1}^{|\mathcal{V}|} y_i^{(t)} \log \hat{y}_i^{(t)}$$

- Conventionally, linguists use **perplexity** to measure the performance of their model, which is just the inverse of the predicted probability of the actual word. That is,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{\hat{p}(x_{t+1}^{pred} = x_{t+1} | x_t, \dots, x_1)} = \frac{1}{\sum_{j=1}^{|\mathcal{V}|} y_j^{(t)} \cdot \hat{y}_j^{(t)}}$$

And since $\mathbf{y}^{(t)}$ is one-hot,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{e^{-CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}} = e^{CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}$$

Therefore, minimizing CE = minimizing perplexity

RNN word based model

Tom Mikolov et al. 2010

- We will try to minimize cost entropy as usual. Namely,

$$CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = - \sum_{i=1}^{|\mathcal{V}|} y_i^{(t)} \log \hat{y}_i^{(t)}$$

- Conventionally, linguists use **perplexity** to measure the performance of their model, which is just the inverse of the predicted probability of the actual word. That is,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{\hat{p}(x_{t+1}^{pred} = x_{t+1} | x_t, \dots, x_1)} = \frac{1}{\sum_{j=1}^{|\mathcal{V}|} y_j^{(t)} \cdot \hat{y}_j^{(t)}}$$

And since $\mathbf{y}^{(t)}$ is one-hot,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{e^{-CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}} = e^{CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}$$

Therefore, minimizing CE = minimizing perplexity

RNN word based model

Tom Mikolov et al. 2010

- We will try to minimize cost entropy as usual. Namely,

$$CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = - \sum_{i=1}^{|\mathcal{V}|} y_i^{(t)} \log \hat{y}_i^{(t)}$$

- Conventionally, linguists use **perplexity** to measure the performance of their model, which is just the inverse of the predicted probability of the actual word. That is,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{\hat{p}(x_{t+1}^{pred} = x_{t+1} | x_t, \dots, x_1)} = \frac{1}{\sum_{j=1}^{|\mathcal{V}|} y_j^{(t)} \cdot \hat{y}_j^{(t)}}$$

And since $\mathbf{y}^{(t)}$ is one-hot,

$$PP^{(t)}(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)}) = \frac{1}{e^{-CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}} = e^{CE(\mathbf{y}^{(t)}, \hat{\mathbf{y}}^{(t)})}$$

Therefore, minimizing CE = minimizing perplexity

Examples

Tom Mikolov et al. 2010

Embedding dim: $d = 50$; Hidden Layer dim: $D_h = 100$

- **The meaning of life is** now.
- **The meaning of life is** myself, that I was going in was looking and Norman Lollards
- **The meaning of life is** just an demons it is, it directed their invisible Joan, at the public-house.
- **The meaning of life is** King Ada to it, something duty out Mr. Dick, who Richard only back for your her dress; Oliver is give by subdued at the old in deplorably meditative under the window, was the concessions

Demo

Name Entity Recognition (NER)

- Goal: try to predict whether a given word in a sentence is a name and its category
 - Person (**PER**)
 - Organization (**ORG**)
 - Location (**LOC**)
 - Miscellaneous (**MISC**)
- For example,
 - **John** lives in **Oklahoma** and studies at the **University of Oklahoma**
 - The **Republicans** will repeal the **Affordable Care Act**

Name Entity Recognition (NER)

See also <http://nlp.stanford.edu/software/CRF-NER.html#Models>

- To get the context of the input word, we may include its neighbors as input

$$\tilde{\mathbf{x}}^{(t)} = [\mathbf{x}^{(t-k)}, \dots, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t+k)}]L \in \mathbb{R}^{3d}$$

- For a simplest model,

$$\mathbf{h} = \tanh(\tilde{\mathbf{x}}^{(t)}W + \mathbf{b}_1)$$

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}U + \mathbf{b}_2)$$

- Let $k = 1$, $d = 50$, $D_h = 100$

Name Entity Recognition (NER)

See also <http://nlp.stanford.edu/software/CRF-NER.html#Models>

- To get the context of the input word, we may include its neighbors as input

$$\tilde{\mathbf{x}}^{(t)} = [\mathbf{x}^{(t-k)}, \dots, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t+k)}]L \in \mathbb{R}^{3d}$$

- For a simplest model,

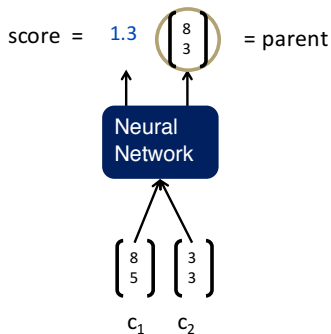
$$\mathbf{h} = \tanh(\tilde{\mathbf{x}}^{(t)} W + \mathbf{b}_1)$$

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}U + \mathbf{b}_2)$$

- Let $k = 1$, $d = 50$, $D_h = 100$

Demo

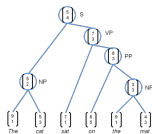
Recursive Neural Network Definition



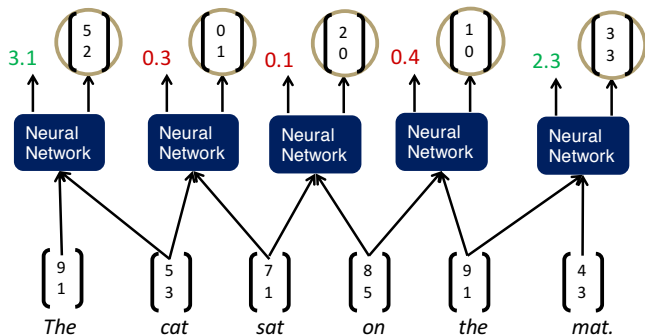
$$\text{score} = U^T p$$

$$p = \tanh(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b),$$

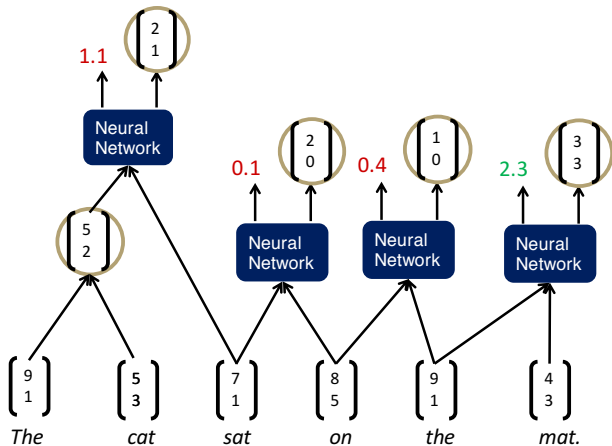
Same W parameters at all nodes of the tree



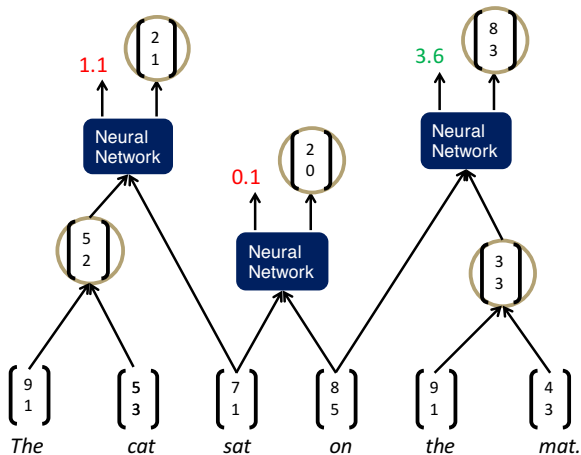
Parsing a sentence with an RNN



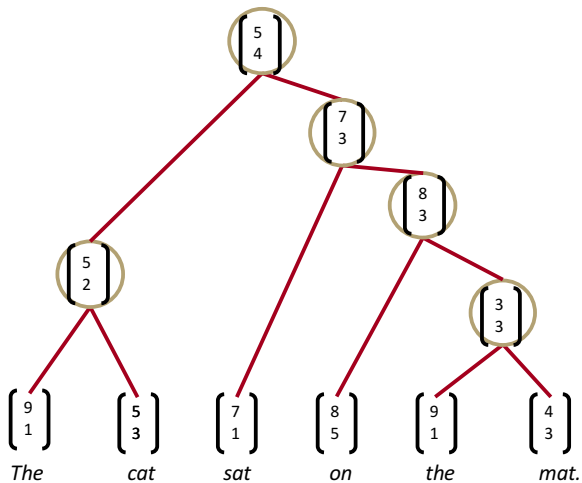
Parsing a sentence



Parsing a sentence



Parsing a sentence



Parsing sentence optimizing overall score

Max-Margin Framework - Details

- The score of a tree is computed by the sum of the parsing decision scores at each node:

$$s(x, y) = \sum_{n \in \text{nodes}(y)} s_n$$



Paraphrase Detection

Pollack said the plaintiffs failed to show that Merrill and Blodgett directly caused their losses

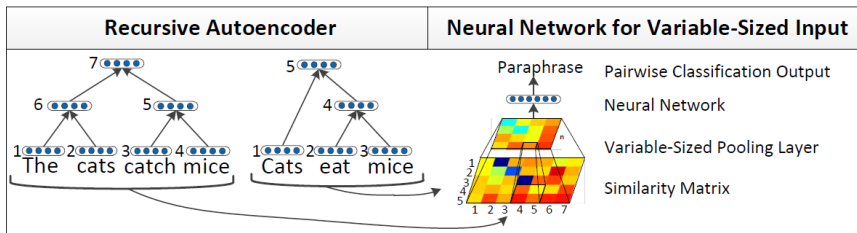
Basically , the plaintiffs did not show that omissions in Merrill's research caused the claimed losses

The initial report was made to Modesto Police December 28

It stems from a Modesto police report

RNNs for Paraphrase Detection

Unsupervised RNNs and a pair-wise sentence comparison of nodes in parsed trees (Socher et al., NIPS 2011)









RNNs for Paraphrase Detection

Experiments on Microsoft Research Paraphrase Corpus
(Dolan et al. 2004)

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



RNNs for Paraphrase Detection

L	Pr	Sentences	Sim.Mat.
P	0.95	(1) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion - Australian football - as the world champion relaxed before his Wimbledon title defence (2) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion-Australian rules football-as the world champion relaxed ahead of his Wimbledon defence	
P	0.82	(1) The lies and deceptions from Saddam have been well documented over 12 years (2) It has been well documented over 12 years of lies and deception from Saddam	
P	0.67	(1) Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses (2) Basically , the plaintiffs did not show that omissions in Merrill's research caused the claimed losses	
N	0.49	(1) Prof Sally Baldwin, 63, from York, fell into a cavity which opened up when the structure collapsed at Tiburtina station, Italian railway officials said (2) Sally Baldwin, from York, was killed instantly when a walkway collapsed and she fell into the machinery at Tiburtina station	
N	0.44	(1) Bremer, 61, is a onetime assistant to former Secretaries of State William P. Rogers and Henry Kissinger and was ambassador-at-large for counterterrorism from 1986 to 1989 (2) Bremer, 61, is a former assistant to former Secretaries of State William P. Rogers and Henry Kissinger	
N 10	0.11	(1) The initial report was made to Modesto Police December 28 (2) It stems from a Modesto police report	

Conclusions

- Briefly introduced NLP problems
 - Language generative model
 - Name entity recognition
 - Paraphrase detection (and parsing)
- Looked into another type of RNNs (recursive neural networks).
Not to be confused with recurrent neural networks

Conclusions

- Briefly introduced NLP problems
 - Language generative model
 - Name entity recognition
 - Paraphrase detection (and parsing)
- Looked into another type of RNNs (recursive neural networks).
Not to be confused with recurrent neural networks

Presentation continuing next week!

Date	Student	Package
3/3	Aakash Soubhi	Tensorflow Tensorflow
3/10	Ahmad A Tamer	Theano Theano
3/23	Ahmad M Obada	Keras Keras
3/30	Muhanad Siraj	Caffe Caffe
4/7	Dong Varun	Torch Lasagne
4/14	Naim	MatConvNet