Natural Language Understanding

Lecture 14: Recursive Autoencoders

Mirella Lapata

School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

March 17, 2016

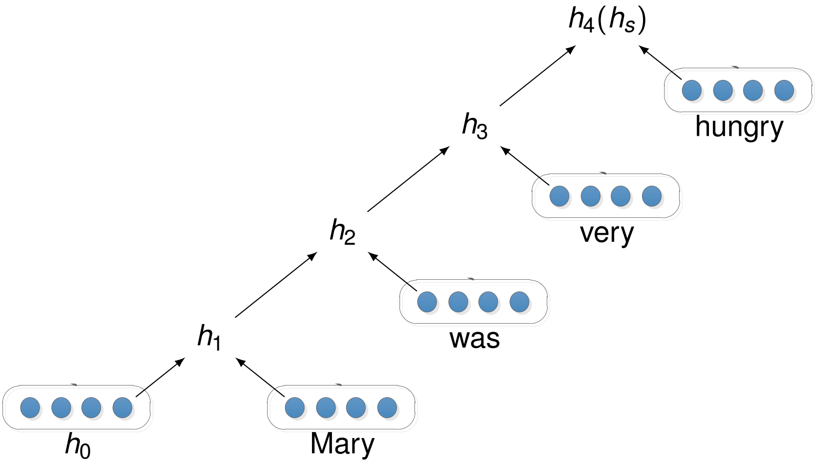
Outline

1. Introduction
2. Recursive Autoencoders
   - Details
   - Reconstruction Error
   - Example
   - In Practice
   - Without Binary Trees
   - Semi-supervised

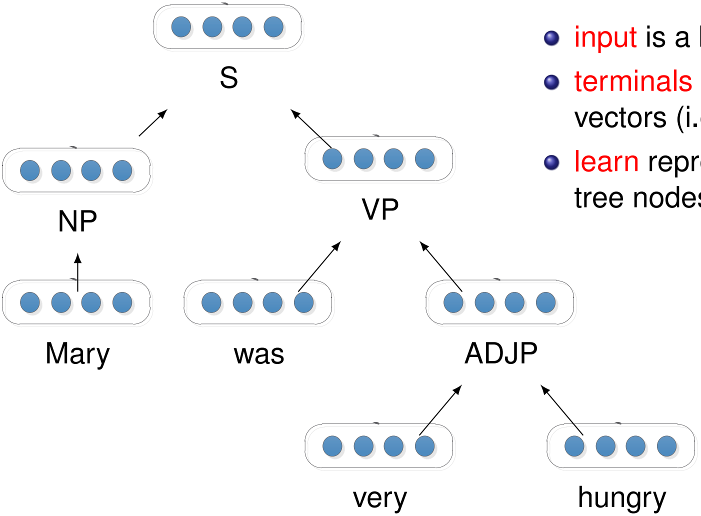
Recurrent Neural Networks

$$\begin{array}{ccccccc}
\begin{bmatrix}1.0\\3.5\end{bmatrix} & \longrightarrow & \begin{bmatrix}1.0\\5.0\end{bmatrix} & \longrightarrow & \begin{bmatrix}5.5\\6.1\end{bmatrix} & \longrightarrow & \begin{bmatrix}4.5\\3.8\end{bmatrix} \\
\Big\uparrow & & \Big\uparrow & & \Big\uparrow & & \Big\uparrow \\
\begin{bmatrix}0.4\\0.3\end{bmatrix} & \longrightarrow & \begin{bmatrix}2.1\\2.3\end{bmatrix} & \longrightarrow & \begin{bmatrix}7\\7\end{bmatrix} & \longrightarrow & \begin{bmatrix}4.0\\4.5\end{bmatrix} \\
\text{Mary} & & \text{was} & & \text{very} & & \text{hungry}
\end{array}$$

Recurrent Neural Networks



Recursive Autoencoders



- input is a binary tree
- terminals represented by vectors (i.e., embeddings)
- learn representations for tree nodes.

Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are syntactically combined.

Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are syntactically combined.

Lakoff (1977): the meaning of the whole is a greater than the meaning of the parts.

Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are syntactically combined.

Lakoff (1977): the meaning of the whole is a greater than the meaning of the parts.

Frege (1884): never ask the meaning of a word in isolation but only in the context of a statement.

Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are syntactically combined.

Lakoff (1977): the meaning of the whole is a greater than the meaning of the parts.

Frege (1884): never ask the meaning of a word in isolation but only in the context of a statement.

Pinker (1994): composition of simple elements must allow the construction of novel meanings which go beyond those of the individual elements.

Recursive Definition of Meaning

Let node k have have children i and j, whose meanings are x_i and x_j. The meaning of node k is:

$$y_k = f(W[x_i; x_j] + b)$$

- W and b are parameters to be learned
- $[x_i; x_j]$ denotes vector x_i concatenated vertically with vector x_j
- therefore W is a matrix in $\mathbb{R}^{d \times 2d}$, b is a bias term, a vector in $\mathbb{R}^d$
- function $f()$ is sigmoid or tahn

How can we train this model in a supervised fashion?

Recursive Definition of Meaning

How can we train this model in a supervised fashion?

$$y_k = f(W[x_i; x_j] + b)$$

- We would need a target value t for the meaning y_r of the whole sentence (r stands for root)
- Then we could define a loss function for E, e.g., square loss $E = (t - y_r)^2$, train the parameters W and b to minimize it
- Compute the gradients $\frac{\partial E}{\partial W}$, $\frac{\partial E}{\partial b}$ using any gradient descent method
- Use SGD, optimize W and b based on one sentence at a time.
- Define error for training set (sum of the errors for each sentence)

Recursive Definition of Meaning

How can we train this model in a supervised fashion?

$$y_k = f(W[x_i; x_j] + b)$$

- We would need a target value t for the meaning y_r of the whole sentence (r stands for root)
- Then we could define a loss function for E, e.g., square loss $E = (t - y_r)^2$, train the parameters W and b to minimize it
- Compute the gradients $\frac{\partial E}{\partial W}$, $\frac{\partial E}{\partial b}$ using any gradient descent method
- Use SGD, optimize W and b based on one sentence at a time.
- Define error for training set (sum of the errors for each sentence)

Recursive Definition of Meaning

How can we train this model in a supervised fashion?

$$y_k = f(W[x_i; x_j] + b)$$

- We would need a target value t for the meaning y_r of the whole sentence (r stands for root)
- Then we could define a loss function for E, e.g., square loss $E = (t - y_r)^2$, train the parameters W and b to minimize it
- Compute the gradients $\frac{\partial E}{\partial W}$, $\frac{\partial E}{\partial b}$ using any gradient descent method
- Use SGD, optimize W and b based on one sentence at a time.
- Define error for training set (sum of the errors for each sentence)

Autoencoders

But we do not know what the target meaning t should be!

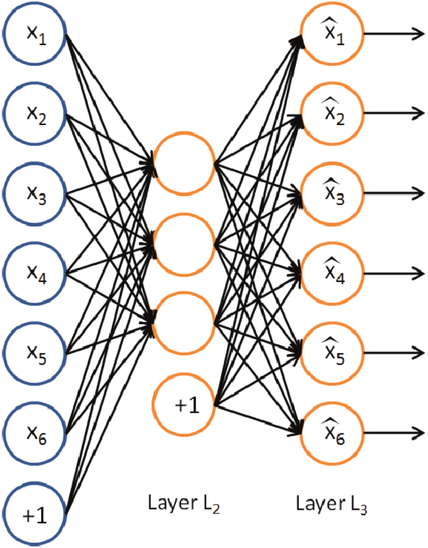
Autoencoders

But we do not know what the target meaning t should be!

Autoencoders: goal of learning is to reconstruct the input!

- Learns function $h_{W,b}(x) \approx x$
- Limit on the number of hidden units
- Learns compressed representation of input
- Can also impose sparsity constraints
- Can be stacked to form highly non-linear representations

Autoencoders



Takes input $\mathbf{x} \in [0,1]^d$, maps it to hidden representation $\mathbf{y} \in [0,1]^{d'}$ through

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Where f is a non-linearity such as the sigmoid.

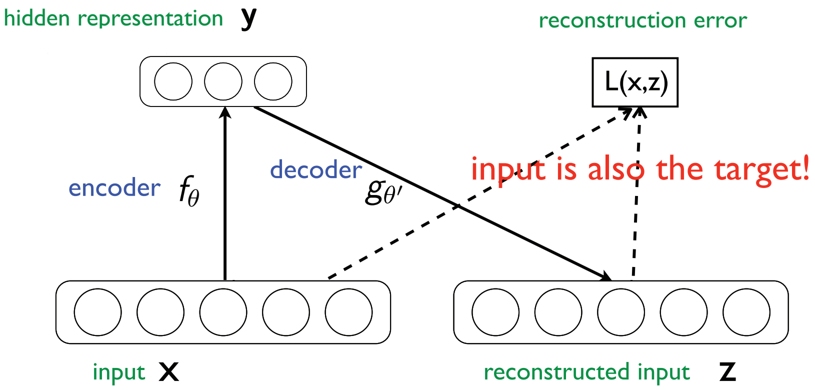
$\mathbf{y}$ is then mapped back (with a decoder) into $\mathbf{z}$:

$$\mathbf{z} = f(\mathbf{W}'\mathbf{y} + \mathbf{b}')$$

$\mathbf{z}$ is a prediction of $\mathbf{x}$, given $\mathbf{y}$

Autoencoders

Unsupervised learning: no explicit target t
Goal: learn lower-dimensional representation



Autoencoders

Parameters optimized so that average reconstruction error is minimized (can be measured in many ways).

- squared error $L(\mathbf{x}, \mathbf{z}) = ||\mathbf{x} - \mathbf{z}||^2$
- cross-entropy of reconstruction:

$$L_H(\mathbf{x}, \mathbf{z}) = -\sum_{k=1}^{d} [\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{z}_k)]$$

- The hope is that **y** is a distributed representation that captures the coordinates along the main factors of variation in the data.
- With one linear hidden layer and mean squared error criterion, k hidden units $\approx$ k principal components.

Recursive Autoencoders

meaning at node k $\quad y_k = f(W[x_i; x_j] + b)$

Recursive Autoencoders

meaning at node k $\quad y_k = f(W[x_i; x_j] + b)$

reconstructions of inputs x_i *and* x_j $\quad [z_i; z_j] = Uy_k + c$

Recursive Autoencoders

meaning at node k $\quad y_k = f(W[x_i; x_j] + b)$

reconstructions of inputs x_i *and* x_j $\quad [z_i; z_j] = Uy_k + c$

- U is a matrix in $\mathbb{R}^{2d \times d}$ and c is a vector in $\mathbb{R}^{2d}$
- z_i and z_j are approximate reconstructions of the inputs x_i and x_j
- U and c are additional parameters to be trained to maximize the accuracy of reconstructions.

Recursive Autoencoders

meaning at node k $\quad y_k = f(W[x_i; x_j] + b)$

reconstructions of inputs x_i *and* x_j $\quad [z_i; z_j] = Uy_k + c$

- U is a matrix in $\mathbb{R}^{2d \times d}$ and c is a vector in $\mathbb{R}^{2d}$
- z_i and z_j are approximate reconstructions of the inputs x_i and x_j
- U and c are additional parameters to be trained to maximize the accuracy of reconstructions.
- Specifically, the square loss at the node k is:

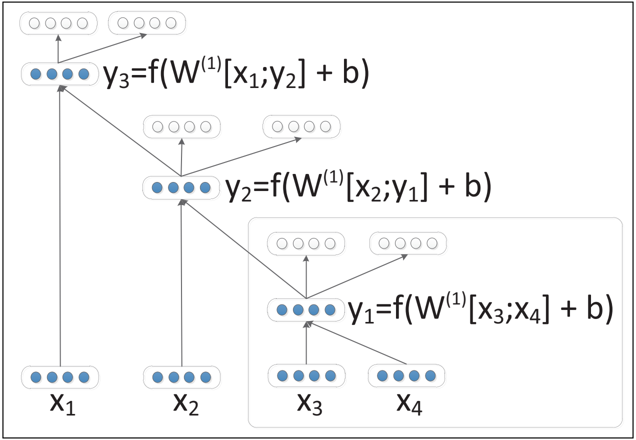
$$
\begin{aligned}
E_{rec} &= \tfrac{1}{2}||[x_i; x_j] - [z_i; z_j]||^2 \\
&= \tfrac{1}{2}||[x_i; x_j] - Uf(W[x_i; x_j] + b) - c||^2
\end{aligned}
$$

Recursive Autoencoders

meaning at node k	$y_k = f(W[x_i; x_j] + b)$
reconstructions of inputs x_i *and* x_j	$[z_i; z_j] = Uy_k + c$
reconstruction error	$E_{rec} = \frac{1}{2}\|\|[x_i; x_j] - [z_i; z_j]\|\|^2$

- The error for a whole tree is the sum of the errors at all the non-leaf nodes of the tree.
- Gradient methods can be used to learn W, b, U, and c, with no training labels provided from the outside.

Recursive Autoencoders



- word vectors $x = (x_1 \dots x_n)$; binary tree structure
- $(y_1 \rightarrow x_3 x_4)$, $(y_2 \rightarrow x_2 y_1)$, $(y_3 \rightarrow x_1 y_2)$
- hidden representations y_i same dimensions as x_i

Recursive Autoencoders

- Avoid ending up with all meanings equal to zero (it would give zero error at all nodes whose children are not leaf nodes).
- Enforce all meaning vectors to have unit length:

$$y_k = \frac{f(W[x_i; x_j] + b)}{||f(W[x_i; x_j] + b)||}$$

- Difficult to reconstruct accurately the meanings of longer phrases.
- The definition of loss for node k is changed to be weighted:

$$E_{rec}(k) = \frac{n_i}{n_i + n_j}||x_i - z_i||^2 + \frac{n_j}{n_i + n_j}||x_j - z_j||^2$$

z_i and z_j are reconstructions; n_i and n_j are the number of words covered by nodes i and j

Recursive Autoencoders

- Avoid ending up with all meanings equal to zero (it would give zero error at all nodes whose children are not leaf nodes).
- Enforce all meaning vectors to have unit length:

$$y_k = \frac{f(W[x_i; x_j] + b)}{||f(W[x_i; x_j] + b)||}$$

- Difficult to reconstruct accurately the meanings of longer phrases.
- The definition of loss for node k is changed to be weighted:

$$E_{rec}(k) = \frac{n_i}{n_i + n_j}||x_i - z_i||^2 + \frac{n_j}{n_i + n_j}||x_j - z_j||^2$$

z_i and z_j are reconstructions; n_i and n_j are the number of words covered by nodes i and j

Selecting a Tree Structure

The reconstruction error for a single leaf node:

$$E_{rec}(k) = \frac{n_i}{n_i + n_j}||x_i - z_i||^2 + \frac{n_j}{n_i + n_j}||x_j - z_j||^2$$

The reconstruction error for a whole tree:

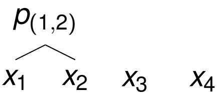
$$\sum_{k \in T} E_{rec}(k)$$

For sentence of length n, there is exponential number of possible trees!

Will use greedy algorithm to find good but not necessarily optimal tree.

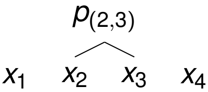
Selecting a Tree Structure

- Consider $n-1$ pairs of consecutive words
- Evaluate reconstruction error for each pair



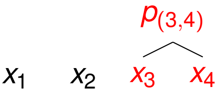
Selecting a Tree Structure

- Consider $n - 1$ pairs of consecutive words
- Evaluate reconstruction error for each pair



Selecting a Tree Structure

- Consider $n - 1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error



Selecting a Tree Structure

- Consider $n - 1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error
- Consider the remaining feasible pairs and new possible pairs on top of the first selected pair.

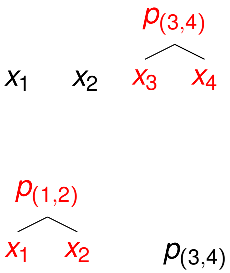
$p_{(3,4)}$

$x_1 \quad x_2 \quad x_3 \quad x_4$

$x_1 \quad x_2 \quad p_{(3,4)}$

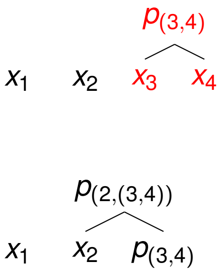
Selecting a Tree Structure

- Consider $n - 1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error
- Consider the remaining feasible pairs and new possible pairs on top of the first selected pair.



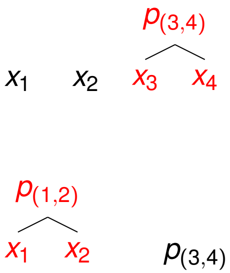
Selecting a Tree Structure

- Consider $n - 1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error
- Consider the remaining feasible pairs and new possible pairs on top of the first selected pair.



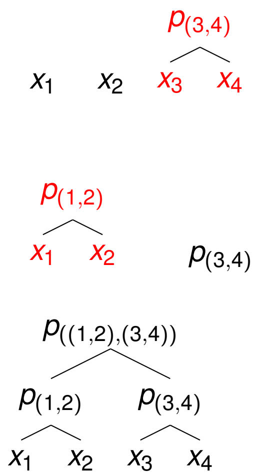
Selecting a Tree Structure

- Consider $n-1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error
- Consider the remaining feasible pairs and new possible pairs on top of the first selected pair.
- Select pair with smallest error



Selecting a Tree Structure

- Consider $n-1$ pairs of consecutive words
- Evaluate reconstruction error for each pair
- Select pair with smallest error
- Consider the remaining feasible pairs and new possible pairs on top of the first selected pair.
- Select pair with smallest error
- Continue until there is only one possible choice to create root



Use Meanings to Predict Labels

Selena Gomez is the raddest	$+1$
She makes Britney Spears sound good	-1

- Each node k of a tree has a meaning vector y_k
- Add a linear model on top of these vectors to predict target values.
- If values are binary, model is a logistic regression classifier.
- If there are three or more discrete values, the model is multinomial or multiclass logistic regression classifier.

Use Meanings to Predict Labels

Vector of predicted probabilities of r label values (V: parameter matrix)

$$\bar{p} = \text{softmax}(Vy_k)$$

Let $\bar{t}$ be binary vector of length r indicating true label value of node k. Squared error of the predictions is $||\bar{t} - \bar{p}||^2$. Alternatively the log loss:

$$E_2(k) = -\sum_{i=1}^{r} t_i \log p_i$$

- We could predict the target value for the entire sentence;
- Instead, predict it for it for all internal nodes (not for leaf nodes).
- Label for the sentence applies to all the phrases of the sentence

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s, t, \theta) + \frac{\lambda}{2} ||\theta||^2$$

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s,t,\theta) + \frac{\lambda}{2} ||\theta||^2$$

- $E(s,t,\theta)$ is the total error for one sentence s with label t

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s,t,\theta) + \frac{\lambda}{2} ||\theta||^2$$

- $E(s,t,\theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s, t >$

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s, t, \theta) + \frac{\lambda}{2} ||\theta||^2$$

- $E(s, t, \theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s, t >$
- $\theta =< W, b, U, c, V >$ is all the parameters of the model

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s,t,\theta) + \frac{\lambda}{2} ||\theta||^2$$

- $E(s,t,\theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s,t >$
- $\theta =< W, b, U, c, V >$ is all the parameters of the model
- λ is the strength of L_2 regularization

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t>\in S} E(s,t,\theta) + \frac{\lambda}{2}||\theta||^2$$

- $E(s,t,\theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s, t >$
- $\theta =< W, b, U, c, V >$ is all the parameters of the model
- λ is the strength of L_2 regularization

$$E(s,t,\theta) = \sum_{k\in T(s)} \alpha E_{rec}(k) + (1-\alpha)E_2(k)$$

Use Meanings to Predict Labels

The objective function to be minimized during learning:

$$J = \frac{1}{m} \sum_{<s,t>\in S} E(s, t, \theta) + \frac{\lambda}{2}||\theta||^2$$

- $E(s, t, \theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s, t >$
- $\theta =< W, b, U, c, V >$ is all the parameters of the model
- λ is the strength of L_2 regularization

$$E(s, t, \theta) = \sum_{k\in T(s)} \alpha E_{rec}(k) + (1 - \alpha)E_2(k)$$

- $T(s)$ set of non-leaf nodes of tree greedily constructed for s

Use Meanings to Predict Labels

The objective function to be minimized during learning:

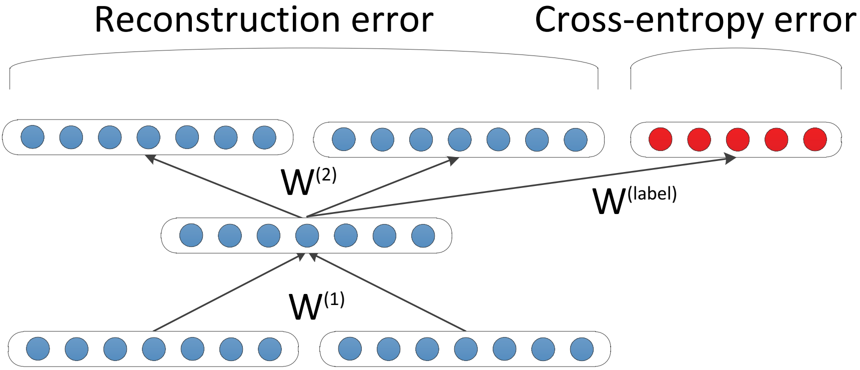
$$J = \frac{1}{m} \sum_{<s,t> \in S} E(s, t, \theta) + \frac{\lambda}{2} ||\theta||^2$$

- $E(s, t, \theta)$ is the total error for one sentence s with label t
- S is collection of m labeled training sentences $< s, t >$
- $\theta =< W, b, U, c, V >$ is all the parameters of the model
- λ is the strength of L_2 regularization

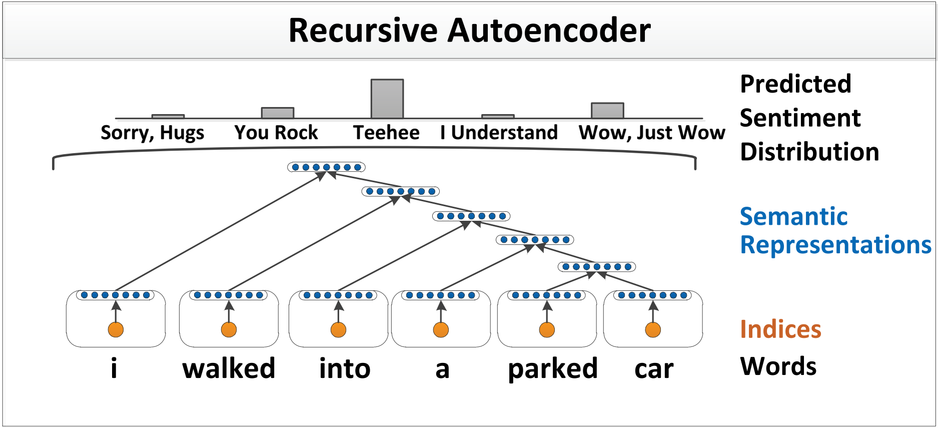
$$E(s, t, \theta) = \sum_{k \in T(s)} \alpha E_{rec}(k) + (1 - \alpha) E_2(k)$$

- $T(s)$ set of non-leaf nodes of tree greedily constructed for s
- α relative importance of reconstruction and label errors.

Use Meanings to Predict Labels



Results on EP Dataset



People anonymously write short personal stories. Once a story is on the site, each user can give a single vote to one of five label categories.

Results on EP Dataset

Method	Accuracy
Random	20.0
Most Frequent	38.1
Baseline 1: Binary BoW	46.4
Baseline 2: Features	47.0
Baseline 3: Word Vectors	45.5
RAE (our method)	**50.1**

Table 1: Accuracy of predicting the class with most votes.

Summary

- Learning compositional representations using recursive autoencoders
- Algorithm can predict sentence level sentiment distributions
- Without using any hand-engineered resources such as sentiment lexica, POS-taggers, or parsers!
- Model learns task specific meaning representations
- Semi-supervised learning is key in learning useful representations.