Introduction

Recurrent Neural Networks

\[
\begin{align*}
\begin{bmatrix} 1.0 \\ 3.5 \end{bmatrix} & \rightarrow \begin{bmatrix} 1.0 \\ 5.0 \end{bmatrix} \\
\begin{bmatrix} 0.4 \\ 0.3 \end{bmatrix} & \rightarrow \begin{bmatrix} 2.1 \\ 2.3 \end{bmatrix}
\end{align*}
\]

Mary was very hungry

\[
\begin{align*}
\begin{bmatrix} 5.5 \\ 6.1 \end{bmatrix} & \rightarrow \begin{bmatrix} 4.5 \\ 3.8 \end{bmatrix} \\
\begin{bmatrix} 7 \\ 7 \end{bmatrix} & \rightarrow \begin{bmatrix} 4.0 \\ 4.5 \end{bmatrix}
\end{align*}
\]

Recurrent Autoencoders

- Details
- Reconstruction Error
- Example
- In Practice
- Without Binary Trees
- Semi-supervised
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.

Lakoff (1977): the meaning of the whole is 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 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 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 tanh

How can we train this model in a supervised fashion?

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 Autoencoders

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: 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

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

\[ y = f(Wx + b) \]

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

\( y \) is then mapped back (with a decoder) into \( z \):

\[ z = f(W' y + b') \]

\( z \) is a prediction of \( x \), given \( y \)
Autoencoders

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

- Parameters optimized so that average reconstruction error is minimized (can be measured in many ways).
  - squared error \( L(x, z) = ||x - z||^2 \)
  - cross-entropy of reconstruction:
    \[
    L_H(x, z) = - \sum_{k=1}^{d} [x_k \log z_k + (1 - x_k) \log(1 - z_k)]
    \]

- 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 \( k \)
\[
y_k = f(W[x_i; x_j] + b)
\]

reconstructions of inputs \( x_i \) and \( x_j \)
\[
[z_i; z_j] = Uy_k + c
\]

- \( U \) 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:
\[
E_{rec} = \frac{1}{2} ||[x_i; x_j] - [z_i; z_j]||^2
\]
\[
= \frac{1}{2} ||[x_i; x_j] - Uf(W[x_i; x_j] + b) - c||^2
\]

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.
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

- 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.

\[
p(2,3) \quad x_1 \quad x_2 \quad x_3 \quad x_4
\]

- \( p(3,4) \)
Recursive Autoencoders Without Binary Trees

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.

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 all internal nodes (not for leaf nodes).
- Label for the sentence applies to all the phrases of the sentence

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 \( <s, t> \)
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
- \( \lambda \) 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
- \( \lambda \) is the strength of \( L_2 \) regularization

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

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

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>20.0</td>
</tr>
<tr>
<td>Most Frequent</td>
<td>38.1</td>
</tr>
<tr>
<td>Baseline 1: Binary BoW</td>
<td>46.4</td>
</tr>
<tr>
<td>Baseline 2: Features</td>
<td>47.0</td>
</tr>
<tr>
<td>Baseline 3: Word Vectors</td>
<td>45.5</td>
</tr>
<tr>
<td>RAE (our method)</td>
<td>50.1</td>
</tr>
</tbody>
</table>

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