Natural Language Processing

Info 159/259
Lecture 4: Text classification 3 (Sept 5, 2017)

David Bamman, UC Berkeley
History of NLP

• Foundational insights, 1940s/1950s
• Two camps (symbolic/stochastic), 1957-1970
• Four paradigms (stochastic, logic-based, NLU, discourse modeling), 1970-1983
• Empiricism and FSM (1983-1993)
• Field comes together (1994-1999)
• Machine learning (2000–today)
• Neural networks (~2014–today)

J&M 2008, ch 1
Neural networks in NLP

• Language modeling [Mikolov et al. 2010]

• Text classification [Kim 2014; Iyyer et al. 2015]

• Syntactic parsing [Chen and Manning 2014, Dyer et al. 2015, Andor et al. 2016]

• CCG super tagging [Lewis and Steedman 2014]


• Dialogue agents [Sordoni et al. 2015, Vinyals and Lee 2015, Ji et al. 2016]

• (for overview, see Goldberg 2017, 1.3.1)
Neural networks

• Discrete, high-dimensional representation of inputs (one-hot vectors) -> low-dimensional “distributed” representations.

• Non-linear interactions of input features

• Multiple “layers” to capture hierarchical structure
Neural network libraries

TensorFlow

Theano

Keras

dy-net
Logistic regression

\[
\hat{y} = \frac{1}{1 + \exp\left(-\sum_{i=1}^{F} x_i \beta_i\right)}
\]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>\beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>1</td>
<td>-0.5</td>
</tr>
<tr>
<td>bad</td>
<td>1</td>
<td>-1.7</td>
</tr>
<tr>
<td>movie</td>
<td>0</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Calculate the derivative of some loss function with respect to parameters we can change, update accordingly to make predictions on training data a little less wrong next time.
Logistic regression

\[ \hat{y} = \frac{1}{1 + \exp \left( - \sum_{i=1}^{F} x_i \beta_i \right)} \]

<table>
<thead>
<tr>
<th>x</th>
<th>β</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>1</td>
</tr>
<tr>
<td>bad</td>
<td>1</td>
</tr>
<tr>
<td>movie</td>
<td>0</td>
</tr>
</tbody>
</table>
Neural networks

• Two core ideas:
  • Non-linear activation functions
  • Multiple layers
*For simplicity, we’re leaving out the bias term, but assume most layers have them as well.*
not
bad
movie
the hidden nodes are completely determined by the input and weights

\[ h_j = f \left( \sum_{i=1}^{F} x_i W_{i,j} \right) \]
\[ h_1 = f \left( \sum_{i=1}^{F} x_i W_{i,1} \right) \]
Activation functions

\[ \sigma(z) = \frac{1}{1 + \exp(-z)} \]
Logistic regression

\[ \hat{y} = \sigma \left( \sum_{i=1}^{F} x_i \beta_i \right) \]

\[ \hat{y} = \frac{1}{1 + \exp \left( - \sum_{i=1}^{F} x_i \beta_i \right)} \]

We can think about logistic regression as a neural network with no hidden layers.
Activation functions

\[ \text{tanh}(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \]
Activation functions

\[
\text{rectifier}(z) = \max(0, z)
\]
\[ h_1 = \sigma \left( \sum_{i=1}^{F} x_i W_{i,1} \right) \]
\[ h_2 = \sigma \left( \sum_{i=1}^{F} x_i W_{i,2} \right) \]
\[ \hat{y} = \sigma [V_1 h_1 + V_2 h_2] \]
we can express \( y \) as a function only of the input \( x \) and the weights \( W \) and \( V \)

\[
\hat{y} = \sigma \left[ V_1 \left( \sigma \left( \sum_{i=1}^{F} x_i W_{i,1} \right) \right) + V_2 \left( \sigma \left( \sum_{i=1}^{F} x_i W_{i,2} \right) \right) \right]
\]

\( W \) and \( V \)
Backpropagation: Given training samples of \(<x, y>\) pairs, we can use stochastic gradient descent to find the values of \(W\) and \(V\) that minimize the loss.

This is hairy, but differentiable
Neural networks are a series of functions chained together.

The loss is another function chained on top.

\[
xW \rightarrow \sigma(xW) \rightarrow \sigma(xW)V \rightarrow \sigma(\sigma(xW)V)
\]

\[
\log(\sigma(\sigma(xW)V))
\]
Chain rule

\[ \frac{\partial}{\partial V} \log (\sigma (\sigma (xW) V)) = \frac{\partial \log (\sigma (\sigma (xW) V))}{\partial \sigma (\sigma (xW) V)} \frac{\partial \sigma (\sigma (xW) V)}{\partial \sigma (xW) V} \frac{\partial \sigma (xW) V}{\partial V} \]

\[ = \frac{\partial \log (\sigma (hV))}{\partial \sigma (hV)} \frac{\partial \sigma (hV)}{\partial hV} \frac{\partial hV}{\partial V} \]

Let’s take the likelihood for a single training example with label \( y = 1 \); we want this value to be as high as possible.
Chain rule

\[
\frac{\partial \log (\sigma (hV))}{\partial \sigma (hV)} \cdot \frac{\partial \sigma (hV)}{\partial hV} \cdot \frac{\partial hV}{\partial V} = \left( \frac{1}{\sigma (hV)} \right) \cdot \sigma (hV) \cdot (1 - \sigma (hV)) \cdot h = (1 - \sigma (hV))h = (1 - \hat{y})h
\]
Neural networks

• Tremendous flexibility on design choices (exchange feature engineering for model engineering)

• Articulate model structure and use the chain rule to derive parameter updates.
Neural network structures

Output one real value
Neural network structures

Multiclass: output 3 values, only one = 1 in training data
Neural network structures

output 3 values, several = 1 in training data
Regularization

- Increasing the number of parameters = increasing the possibility for overfitting to training data
Regularization

- L2 regularization: penalize $W$ and $V$ for being too large

- Dropout: when training on a $<x,y>$ pair, randomly remove some node and weights.

- Early stopping: Stop backpropagation before the training error is too small.
Deeper networks

\[ W_1 \quad W_2 \quad V \]

\[ x_1 \quad x_2 \quad x_3 \]

\[ h_1 \quad h_2 \quad h_2 \quad h_2 \]

\[ y \]
Densely connected layer

\[ h = \sigma(xW) \]
Convolutional networks

- With convolution networks, the same operation is (i.e., the same set of parameters) is applied to different regions of the input
2D Convolution

1D Convolution

convolution $K$

\[
x_{1:4}K = \text{moving average}
\]
Convolutional networks

\[ h_1 = f(I, \text{hated, it}) \]

\[ h_2 = f(\text{it, I, really}) \]

\[ h_3 = f(\text{really, hated, it}) \]

\[ h_1 = \sigma(x_1 W_1 + x_2 W_2 + x_3 W_3) \]

\[ h_2 = \sigma(x_3 W_1 + x_4 W_2 + x_5 W_3) \]

\[ h_3 = \sigma(x_5 W_1 + x_6 W_2 + x_7 W_3) \]
Indicator vector

- Every token is a $V$-dimensional vector (size of the vocab) with a single $1$ identifying the word.

- We’ll get to distributed representations of words in on 9/19.

<table>
<thead>
<tr>
<th>vocab item</th>
<th>indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>aa</td>
<td>0</td>
</tr>
<tr>
<td>aal</td>
<td>0</td>
</tr>
<tr>
<td>aalii</td>
<td>0</td>
</tr>
<tr>
<td>aam</td>
<td>0</td>
</tr>
<tr>
<td>aardvark</td>
<td>1</td>
</tr>
<tr>
<td>aardwolf</td>
<td>0</td>
</tr>
<tr>
<td>aba</td>
<td>0</td>
</tr>
</tbody>
</table>
Convolutional networks

\[ h_1 = \sigma(x_1W_1 + x_2W_2 + x_3W_3) \]
\[ h_2 = \sigma(x_3W_1 + x_4W_2 + x_5W_3) \]
\[ h_3 = \sigma(x_5W_1 + x_6W_2 + x_7W_3) \]
For indicator vectors, we’re just adding these numbers together

\[ h_1 = \sigma(W_{1,x_1^id} + W_{2,x_2^id} + W_{3,x_3^id}) \]

(Where \( x_n^id \) specifies the location of the 1 in the vector — i.e., the vocabulary id)
Pooling

- Down-samples a layer by selecting a single point from some set
- Max-pooling selects the largest value
Convolutional networks

This defines one filter.

convolution  max pooling
\begin{align*}
h_1 &= \sigma(x^\top W) \end{align*}
We can specify multiple filters; each filter is a separate set of parameters to be learned.

\[ h_1 = \sigma(x^\top W) \in \mathbb{R}^4 \]
We can specify multiple filters; each filter is a separate set of parameters to be learned.

\[ h_1 = \sigma(x^\top W) \in \mathbb{R}^4 \]
Convolutional networks

• With max pooling, we select a single number for each filter over all tokens

• (e.g., with 100 filters, the output of max pooling stage = 100-dimensional vector)

• If we specify multiple filters, we can also scope each filter over different window sizes
\[
tanh(x \circ W) \rightarrow \text{max} \rightarrow \sigma(hV)
\]
CNN as important ngram detector

Higher-order ngrams are much more informative than just unigrams (e.g., “i don’t like this movie” [“I”, “don’t”, “like”, “this”, “movie”])

We can think about a CNN as providing a mechanism for detecting important (sequential) ngrams without having the burden of creating them as unique features

<table>
<thead>
<tr>
<th>unique types</th>
<th>50921</th>
</tr>
</thead>
<tbody>
<tr>
<td>unigrams</td>
<td>50921</td>
</tr>
<tr>
<td>bigrams</td>
<td>451,220</td>
</tr>
<tr>
<td>trigrams</td>
<td>910,694</td>
</tr>
<tr>
<td>4-grams</td>
<td>1,074,921</td>
</tr>
</tbody>
</table>

Unique ngrams (1-4) in Cornell movie review dataset
CNN Backprop (V)

\[ L(W, V) = y \log o + (1 - y) \log(1 - o) \]

\[ \frac{\partial L(W, V)}{\partial V} = \frac{\partial A}{\partial V} + \frac{\partial B}{V} = \frac{\partial A}{V} + \frac{\partial B}{V} \]
\[
\frac{\partial A}{\partial V} = \frac{\partial y \log(\sigma(Vh))}{\partial \sigma(Vh)} \times \frac{\partial \sigma(Vh)}{\partial Vh} \times \frac{\partial Vh}{\partial V}
\]
\[
= \frac{y}{\sigma(Vh)} \times \sigma(Vh)(1 - \sigma(Vh)) \times h
\]
\[
= y(1 - \sigma(Vh))h
\]
\[
\frac{\partial B}{\partial V} = \frac{\partial (1 - y) \log (1 - \sigma (Vh))}{\partial (1 - \sigma (Vh))} \times \frac{\partial (1 - \sigma (Vh))}{\partial Vh} \times \frac{\partial Vh}{\partial V}
\]

\[
= \frac{1 - y}{1 - \sigma (Vh)} \times -\sigma (Vh) (1 - \sigma (Vh)) \times h
\]

\[
= -(1 - y) (\sigma (Vh)) h
\]
CNN Backprop (V)

\[
\frac{\partial A + B}{V} = \frac{\partial A}{V} + \frac{\partial B}{V} \\
= y (1 - \sigma (Vh)) h - (1 - y) (\sigma (Vh)) h \\
= (y - \sigma (Vh)) h
\]
• You’ll derive and implement updates for the rest of the parameters in homework 2
Generative vs. Discriminative models

• Generative models specify a joint distribution over the labels and the data. With this you could generate new data

\[ P(x, y) = P(y) \cdot P(x \mid y) \]

• Discriminative models specify the conditional distribution of the label \( y \) given the data \( x \). These models focus on how to discriminate between the classes

\[ P(y \mid x) \]
259 project proposal
due 9/26

• Final project involving 1 or 2 students involving natural language processing -- either focusing on core NLP methods or using NLP in support of an empirical research question.

• Proposal (2 pages):
  • outline the work you’re going to undertake
  • motivate its rationale as an interesting question worth asking
  • assess its potential to contribute new knowledge by situating it within related literature in the scientific community. (cite 5 relevant sources)
  • who is the team and what are each of your responsibilities (everyone gets the same grade)
Thursday

- Read Hovy and Spruit (2016) and come prepared to discuss!