Structure your model

CS 20SI: TensorFlow for Deep Learning Research
Lecture 4
1/25/2017
Assignment 1 is out! (due 1/31)

Questions?
Agenda

Overall structure of a model in TensorFlow

word2vec

Name scope

Embedding visualization

Interactive Coding!
Overall structure of a model in TensorFlow
Phase 1: Assemble graph

1. Define placeholders for input and output
2. Define the weights
3. Define the inference model
4. Define loss function
5. Define optimizer
Phase 2: Compute

Training loop

- Initialize model parameters
- Input training data
- Execute inference model on training data
- Compute loss
- Adjust model parameters

Graph from the book “TensorFlow for Machine Intelligence”
Word Embedding

Capture the semantic relationships between words
Word Embedding
Live visualization
Count vs Predict
### Counting

- Example corpus:
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

<table>
<thead>
<tr>
<th>counts</th>
<th>I</th>
<th>like</th>
<th>enjoy</th>
<th>deep</th>
<th>learning</th>
<th>NLP</th>
<th>flying</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>like</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>enjoy</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>deep</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NLP</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>flying</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Predicting

Graph by deeplearning4j.org
Implementing word2vec skip-gram
Softmax vs Sample-based Approaches
Softmax

\[ p(o|c) = \frac{\exp(u_o^T V_c)}{\sum_{w=1}^{V} \exp(u_w^T V_c)} \]

Computationally expensive
Sample-based Approaches

Negative Sampling is a simplified version of Noise Contrastive Estimation.
Sample-based Approaches

NCE guarantees approximation to softmax

Negative Sampling doesn’t

See lecture note for mathy stuff
Embedding Lookup

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\times
\begin{bmatrix}
17 & 24 & 1 \\
23 & 5 & 7 \\
4 & 6 & 13 \\
10 & 12 & 19 \\
11 & 18 & 25
\end{bmatrix}
= \begin{bmatrix}
10 & 12 & 19
\end{bmatrix}
\]
Embedding Lookup

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\times
\begin{bmatrix}
17 & 24 & 1 \\
23 & 5 & 7 \\
4 & 6 & 13 \\
10 & 12 & 19 \\
11 & 18 & 25 \\
\end{bmatrix}
= \begin{bmatrix}
10 & 12 & 19
\end{bmatrix}
\]

\[
tf.nn.embedding_lookup(params, ids, partition_strategy='mod', name=None, validate_indices=True, max_norm=None)
\]

Illustration by Chris McCormick
NCE Loss

tf.nn.nce_loss(weights, biases, labels, inputs, num_sampled, num_classes, ...)

Illustration by Chris McCormick
Let’s write some code

Get `process_data.py` and `04_word2vec_starter.py` from GitHub
(in examples)
Name scope

Group nodes together

with tf.name_scope(name)
Next class

Manage experiments

Example: word2vec

Feedback: huyenn@stanford.edu

Thanks!