Neural Network-based Language Model for Conversational Telephone Speech Recognition

Graeme W. Blackwood

Supervised by Professor Phil Woodland

Based on papers by Schwenk and Gauvain 2002 - 2004
Introduction & Motivation

• Main limitation of $n$-grams = poor generalization
  – Discrete word indices means probability distributions are not smooth
  – This problem is especially acute when training data is limited

• Advantage of neural network approach
  – Train network to classify continuous vector space representation of $n$-gram history
  – Smooth distribution allows better generalisation to unseen testing data

• Neural network issue – very long training time
  – Requires integration with machine-specific optimized linear algebra libraries and gradient descent optimizations
Neural Network Architecture

• Network inputs are the word indices of the \( n \)-gram context
  \[
  h_j = w_{j-n+1}, w_{j-n+2}, \ldots, w_{j-1}
  \]

• Network outputs are the posterior probabilities of all words in the vocabulary
  \[
  P(w_j = i | h_j), \quad \forall i \in [1 \ldots N]
  \]
Neural Network Architecture

\[ P(w_j = 1 \mid h_j) \]
\[ P(w_j = i \mid h_j) \]
\[ P(w_j = n \mid h_j) \]
Neural Network Training

- Map context to continuous vector space representation
- Train as probability estimator using softmax normalization
- Standard error back propagation & gradient descent
- Use cross entropy as objective function during training

\[ E = \sum_{i=0}^{N-1} d_i \log p_i \]

- Minimizes perplexity of the training data

\[ H = - \sum_{j=0}^{N} \log_2 P(w_j = i | h_j) \]

\[ PP = 2^H \]
Baseline: Restricted Vocabularies

- Evaluated network for a range of restricted vocabulary sizes
- 50, 100, 500, 1000, and 2000 words
- Aim is to demonstrate improved perplexity over the $n$-gram approach
- All out-of-vocabulary words mapped to an OOV input / output
Restricted Vocabularies: Results

Training and testing perplexities for Switchboard I
(~3M words, fixed topology P=24, H=48)

Training

Testing

[Bar charts showing training and testing perplexities for different vocabulary sizes, comparing N-gram and Neural Network models.]
Shortlist Integration

• Computing probabilities for full vocabulary very expensive

• Many words rarely requested in practice

• Define shortlist of frequent words and use neural network for shortlist words only

\[
P(w_j \mid h_j) = \begin{cases} 
P_N(w_j \mid h_j) \cdot P_S(h_j) & \text{if } w_j \in \text{shortlist} \\ 
P_B(w_j \mid h_j) & \text{otherwise} \end{cases}
\]

• Normalization

\[
P_S(h_j) = \sum_{w \in \text{shortlist}} P_B(w \mid h_j)
\]
Shortlist Integration: Results

- Topology: ~26000 inputs, 1001 outputs
- Compare directly with full vocabulary $n$-gram language model perplexity of 79.71
- Did not show expected improvements in perplexity

<table>
<thead>
<tr>
<th>$P_N(w_j \mid h_j)$</th>
<th>$P_B(w_j \mid h_j)$</th>
<th>Hidden Layer Size</th>
<th>Training Perplexity</th>
<th>Testing Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.29%</td>
<td>9.71%</td>
<td>64</td>
<td>48.84</td>
<td>87.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>128</td>
<td>46.81</td>
<td>85.92</td>
</tr>
</tbody>
</table>

Switchboard I, 1000 word shortlist, P=48
Extension: Class Models

- This extension applied the neural network language model to a class-based word prediction task.
- Class-based $n$-grams cope better with data sparsity.
- Word probability computed as product of class $n$-gram probability and class conditional unigram probability:

\[ P(w_j \mid w_{j-n+1}, w_{j-n+2}, \ldots, w_{j-1}) = P(c_j \mid c_{j-n+1}, c_{j-n+2}, \ldots, c_{j-1}) \cdot P(w_j \mid c_j) \]

- Two possible strategies:
  - Train network to predict word outputs directly.
  - Train network to predict class outputs & combine with class conditional probabilities.
Class Model Architecture

Map to Class IDs and Unigram Probability

\[
P(w_j | h_j) = P(w_j | c_j = t) \cdot P(c_j = t | h_j)
\]
Class Model Results

Testing perplexity for Cellular 1, ~220K words

<table>
<thead>
<tr>
<th>Vocabulary Size</th>
<th>Classes</th>
<th>Testing Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N-gram</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>57.53</td>
</tr>
<tr>
<td>2000</td>
<td>150</td>
<td>67.03</td>
</tr>
<tr>
<td>Full</td>
<td>500</td>
<td>81.30</td>
</tr>
</tbody>
</table>

Testing perplexity for Switchboard I, ~3M words

<table>
<thead>
<tr>
<th>Vocabulary Size</th>
<th>Classes</th>
<th>Testing Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N-gram</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>59.38</td>
</tr>
<tr>
<td>2000</td>
<td>150</td>
<td>65.41</td>
</tr>
<tr>
<td>Full</td>
<td>500</td>
<td>85.08</td>
</tr>
</tbody>
</table>
Conclusions

• Demonstrated consistently improved perplexity for restricted vocabularies
  – Relative performance of neural network declines as size of vocabulary increases
• Training time is a real issue – very important to ensure convergence is as fast as possible
• Class-based models of limited utility – performance improvements possible only when training data is limited
Further Work

• Lattice rescoring or full integration with ASR system
• Reduce training time: bunch-mode and 2nd order gradient descent methods
  – E.g. Quickprop
• Training with larger corpora + longer training times
  – Combined Switchboard / Cellular corpora