Recap: Neural Networks for Classification

\[ P(y|x) = \text{softmax}(Wg(Vf(x))) \]

- \( f(x) \): \( n \) features \( d \times n \) matrix
- \( g \): nonlinearity (tanh, relu, ...)
- \( z \): \( d \) hidden units
- \( W \): \( num\_classes \times d \) matrix
- \( P(y|x) \): \( num\_classes \) probs

Announcements

- Fairness response due today
- A2 due in 9 days
Word Embeddings

- Currently we think of words as “one-hot” vectors
  \[ \text{the} = v_{\text{the}} = [1, 0, 0, 0, 0, 0, \ldots] \]
  \[ \text{good} = v_{\text{good}} = [0, 0, 0, 1, 0, 0, \ldots] \]
  \[ \text{great} = v_{\text{great}} = [0, 0, 0, 0, 0, 1, \ldots] \]
- \text{good} and \text{great} seem as dissimilar as \text{good} and \text{the}
  \[ \text{the movie was great} = v_{\text{the}} + v_{\text{movie}} + v_{\text{was}} + v_{\text{great}} \]
- Neural networks are built to learn sophisticated nonlinear functions of continuous inputs; our inputs are discrete and high-dimensional

Deep Averaging Networks

- Deep Averaging Networks: feedforward neural network on average of word embeddings from input

\[ h_2 = f(W_2 \cdot h_1 + b_2) \]
\[ h_1 = f(W_1 \cdot av + b_1) \]
\[ av = \frac{1}{4} \sum_{i=1}^{4} c_i \]

Iyyer et al. (2015)
**Sentiment Analysis**

<table>
<thead>
<tr>
<th>Model</th>
<th>RT</th>
<th>SST Fine</th>
<th>SST Bin</th>
<th>IMDB</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN-ROOT</td>
<td>64.9</td>
<td>85.7</td>
<td>—</td>
<td>31</td>
<td>—</td>
</tr>
<tr>
<td>DAN-RAND</td>
<td>77.3</td>
<td>83.2</td>
<td>88.8</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>DAN</td>
<td>80.3</td>
<td>86.3</td>
<td>89.4</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>NBO-RAND</td>
<td>76.2</td>
<td>81.4</td>
<td>88.9</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>NBO</td>
<td>79.0</td>
<td>83.6</td>
<td>89.0</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>BINB</td>
<td>—</td>
<td>83.1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>NBSVM-bi</td>
<td>79.4</td>
<td>—</td>
<td>91.2</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

- Iyyer et al. (2015)
- Wang and Manning (2012)
- Kim (2014)

---

**Word Embeddings in PyTorch**

- torch.nn.Embedding: maps vector of indices to matrix of word vectors
- 
  Predator is a masterpiece
  
  
  1820 24 1 2047

- $n$ indices => $n \times d$ matrix of $d$-dimensional word embeddings
- $b \times n$ indices => $b \times n \times d$ tensor of $d$-dimensional word embeddings

---

**Word Embeddings**

- J.R. Firth, 1957: “You shall know a word by the company it keeps.”
  
  I watched the movie
  I watched the baby
  The movie inspired me
  The film inspired me

  There was film on the liquid