The perceptron, again

\[ \hat{y}_i = \begin{cases} 
1 & \text{if } \sum_i x_i \beta_i \geq 0 \\
-1 & \text{otherwise}
\end{cases} \]

<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>0.4</td>
</tr>
<tr>
<td>movie</td>
<td>0</td>
<td>1.7</td>
</tr>
</tbody>
</table>
The perceptron, again

\[ y_i = \begin{cases} 1 & \text{if } \sum x_i \beta_i \geq 0 \\ -1 & \text{otherwise} \end{cases} \]

<table>
<thead>
<tr>
<th>x</th>
<th>\beta</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
Input  "Hidden"  Output
Layer
not
bad
movie

<table>
<thead>
<tr>
<th>x</th>
<th>W</th>
<th>V</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.5</td>
<td>4.1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0.4</td>
<td>-0.9</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The hidden nodes are completed 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

\[ \sigma(z) = \frac{1}{1 + \exp(-z)} \]
Activation functions

$$tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$
Activation functions

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

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

This is hairy, but differentiable

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

Output one real value
Neural network structures

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

x1 \rightarrow h1 \rightarrow y

x2 \rightarrow h1 \rightarrow y

x3 \rightarrow h2 \rightarrow y

output 3 values, several = 1 in training data
Deeper networks
Is there an eye in the top left?

Is there an eye in the top right?

Is there a nose in the middle?

Is there a mouth at the bottom?

Is there hair on top?

Is this a face?
Higher order features learned for image recognition
Lee et al. 2009 (ICML)
Autoencoder

- Unsupervised neural network, where $y = x$
- Learns a low-dimensional representation of $x$
Feedforward networks

\[ x_1 \rightarrow h_1 \rightarrow h_2 \rightarrow y \]

\[ x_2 \rightarrow h_1 \]

\[ x_3 \rightarrow h_2 \]
Recurrent networks

- Input (x)
- Hidden layer (h)
- Label (y)
Recurrent networks

\[ x_1 \rightarrow h_1 \rightarrow y \]

\[ x_2 \rightarrow h_1 \rightarrow y \]

\[ x_3 \rightarrow h_1 \rightarrow y \]

\[ v \]

\[ x_1 \rightarrow h_1 \rightarrow y \]

\[ x_2 \rightarrow h_1 \rightarrow y \]

\[ x_3 \rightarrow h_1 \rightarrow y \]

\[ w \]

\[ \text{time step 1} \quad \text{time step 2} \]
Recurrent networks

\[ h_1 = \sigma \left( \sum_{i=1}^{F} W_{i,1}x_i + \sum_{j=1}^{H} U_{j,1}h_j^{t-1} \right) \]

\( h_1 \) and \( h_2 \) are the hidden layers at time step 2.
Recurrence networks

RNNs often have a problem with long-distance dependencies.
LSTMs

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Recurrent networks/LSTMs

<table>
<thead>
<tr>
<th>task</th>
<th>(x)</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>language models</td>
<td>words in sequence</td>
<td>the next word in a sequence</td>
</tr>
<tr>
<td>part of speech tagging</td>
<td>words in sequence</td>
<td>part of speech</td>
</tr>
<tr>
<td>machine translation</td>
<td>words in sequence</td>
<td>translation</td>
</tr>
</tbody>
</table>