Recurrent Neural Networks
Greg Mori - CMPT 419/726

Goodfellow, Bengio, and Courville: Deep Learning textbook
Ch. 10
Sequential Data with Neural Networks

• Sequential input / output
  • Many inputs, many outputs $x_{1:T} \rightarrow y_{1:S}$
    • c.f. object tracking, speech recognition with HMMs; on-line/batch processing
  • One input, many outputs $x \rightarrow y_{1:S}$
    • e.g. image captioning
  • Many inputs, one output $x_{1:T} \rightarrow y$
    • e.g. video classification
Outline

Recurrent Neural Networks

Long Short-Term Memory

Temporal Convolutional Networks

Examples
Outline

Recurrent Neural Networks

Long Short-Term Memory

Temporal Convolutional Networks

Examples
Hidden State

• Basic idea: maintain a state $h_t$
• State at time $t$ depends on input $x_t$ and previous state $h_{t-1}$
• It’s a neural network, so relation is non-linear function of these inputs and some parameters $W$:

$$h_t = f(h_{t-1}, x_t; W)$$

• Parameters $W$ and function $f(\cdot)$ reused at all time steps
Outputs

- Output $y_t$ also depends on the hidden state:

  $$y_t = f(h_t; W_y)$$

- Again, parameters/function reused across time
Outline

Recurrent Neural Networks

Long Short-Term Memory

Temporal Convolutional Networks

Examples
Gradients

- Basic RNN not very effective
- Need many time steps / complex model for challenging tasks
- Gradients in learning are a problem
  - Too large: can be handled with gradient clipping (truncate gradient magnitude)
  - Too small: can be handled with network structures / gating functions (LSTM, GRU)
Long Short-Term Memory

- Hochreiter and Schmidhuber, Neural Computation 1997
  - (Figure from Donohue et al. CVPR 2015)
- **Gating functions** $g(\cdot), f(\cdot), o(\cdot)$, reduce vanishing gradients
Long Short-Term Memory

\[ i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \]  
\[ f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \]  
\[ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \]  
\[ g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \]  
\[ c_t = f_t \odot c_{t-1} + i_t \odot g_t \]  
\[ h_t = o_t \odot \tanh(c_t) \]  

Outline

Recurrent Neural Networks

Long Short-Term Memory

Temporal Convolutional Networks

Examples
Convolutions to Aggregate over Time

- Control history by $d$ (dilation, holes in the filter) and $k$ (width of the filter)
- Causal convolution, only use elements from the past
- Bai, Kolter, Koltun arXiv 2018
Residual (skip) Connections

- Include residual connections to allow long-range modeling and gradient flow
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrent Neural Networks</td>
</tr>
<tr>
<td>Long Short-Term Memory</td>
</tr>
<tr>
<td>Temporal Convolutional Networks</td>
</tr>
<tr>
<td>Examples</td>
</tr>
</tbody>
</table>
Example: Image Captioning

- Karpathy and Fei-Fei, CVPR 2015
Example: Video Description

Example: Machine Translation

### Conclusion

- **Readings:** [http://www.deeplearningbook.org/contents/rnn.html](http://www.deeplearningbook.org/contents/rnn.html)

- **Recurrent neural networks, can model sequential inputs/outputs**
  - Input includes state (output) from previous time
  - Different structures:
    - RNN with multiple inputs/outputs
    - Gated recurrent unit (GRU)
    - Long short-term memory (LSTM)
  - Error gradients back-propagated across entire sequence