Generating text with Recurrent Neural Networks

Ilya Sutskever
James Martens
Geoff Hinton
Recurrent Neural Networks
Recurrent Neural Networks
Backprop

Gradient decay / blowup
A source of the difficulty

- Tiny gradient
A source of the difficulty

- Tiny gradient
A source of the difficulty

- Giant gradient: instability
Hessian-Free optimization

- A practical large-scale 2\textsuperscript{nd} order optimization technique

- It can optimize RNNs
Hessian-Free optimization

- A remarkable 2nd-order optimization technique
- Partially invert the curvature using linear Conjugate Gradient
  - Only requires matrix-vector products
- Use the **exact** Hessian

\[
H \nu = \frac{\nabla L(\theta + \epsilon \nu) - \nabla L(\theta - \epsilon \nu)}{2 \epsilon}
\]
Conjugate Gradient

- Conjugate gradient optimizes quadratic functions
  \[
  \frac{\delta^T B \delta}{2} + g^T \delta
  \]

- Only requires computing \( Bv \) products

- At step \( i \), it finds the optimal solution in
  \[
  \text{span}\{g, Bg, B^2g, ..., B^{i-1}g\}
  \]
  - Converges in \( N \) steps or less
Differences from Quasi-Newton methods

- Quasi-Newton: exact minimization on a very crude quadratic approximation

- Hessian-Free: partial minimization on an extremely rich quadratic approximation
Why is HF better than Nonlinear Conjugate gradient?

- Conjugate gradient strongly assumes that the function is quadratic

- Nonlinear CG is a hack: apply CG as is to a nonlinear function and hope for the best

- In contrast, the HF approach says: make the conditions where CG shines
Applying HF optimization to RNNs

- Essentially a straightforward application of Hessian-free optimization

- But it's important to use structural damping:
  - Normal damping asks the parameters to not change too much
  - Structural damping asks internal variables to not change too much
Structural damping

• Take our quadratic approximation, and add a nonlinear objective that doesn't want the hidden state sequence to change
• Then use a quadratic approximation of this term
  - Must do so for CG to be applicable
• The resulting can be obtained with no extra work!
Character-level language modelling

- RNNs were, until now, impossibly hard to optimize
- Hessian-Free optimization is really powerful and can optimize RNNs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RNN</th>
<th>Memoizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIKI</td>
<td>1.60</td>
<td>1.66</td>
</tr>
<tr>
<td>NYT</td>
<td>1.49</td>
<td>1.48</td>
</tr>
<tr>
<td>ML</td>
<td>1.33</td>
<td>1.31</td>
</tr>
</tbody>
</table>
The 500-timesteps multiplication problem

- Shows that the Hessian-Free optimizer has little problem with Long-Term dependencies

- Cannot be solved without structural damping
Major application

- Train an RNN with 2000 units to predict the next character in Wikipedia