Abstract

- Vanishing long-term gradients is a major issue in RNN training.
- LSTM solves vanishing gradients with memory cells, but 4× as many parameters as an RNN.
- We propose to use high order (Markovian) RNNs (HORNN) with extra connections from multiple previous time steps.
- ASR experiments on English multi-genre broadcast (MGB3) data showed sigmoid and ReLU HORNNs reduced WERs by 6.3% and 4.2% over RNNs.
- HORNNs gave similar WERs to projected LSTMs (LSTMP) by using 20%–50% of the RNN parameters and computation.

Markov Conditional Independence

- A standard RNN layer is defined as $h_t = \text{f}(Wx_t + Uh_{t-1} + b)$.
- Standard HORNN model structures are ReLU: $h_t = \text{f}(Wx_t + Uh_{t-1} + Uh_{t-2} + b)$, Sigmoid: $h_t = \text{f}(Wx_t + Uh_{t-1} + Uh_{t-2} + b)$.
- Sigmoid HORNN requires to add extra $h_{t-2}$, since it suffers from vanishing gradients.
- The gradients of an objective function $F$ w.r.t. $h_t$ is

$$\frac{\partial F}{\partial h_t} = -f\left(\sum_{i=1}^{m} \frac{\partial F}{\partial h_n} W_{ih} + \frac{\partial F}{\partial h_m} U_{mh}\right)$$

where $h_t$ and $x_t$ are the output and input vectors at time $t$; $W$ and $U$ are weight matrices, $b$ is bias vector; $f(\cdot)$ is the activation function.

HORNNs have the 1st-order Markov conditional independence property (detailed proof given in paper). $h_t$ is continuous-valued hidden state; depends only on $h_{t-1}$, $x_t$.

- Property also applies to bidirectional RNNs by viewing $h_{t-1} = \text{f}(W_{ih} h_{t-2} + b)$.
- RNN language models (LM) have std. 1st-order Markov property.

HORNNs for Sigmoid and ReLU Functions

- Vanishing gradients can be solved by relaxing the 1st-order Markov conditional independence constraint.
- Use $n$-order Markov property by including $h_{t-n}$ in $h_t$ calculation.
- To directly access to long-term information in testing.
- To create shortcuts to allow additional long-term information to flow more easily in training.
- The gradients of an objective function $F$ w.r.t. $h_t$ is

$$\frac{\partial F}{\partial h_t} = \frac{\partial F}{\partial h_{t-1}} W_{ih} + \frac{\partial F}{\partial h_{t-2}} U_{mh}$$

that alleviates long-term information vanishing by integrating it explicitly at each time step.

Experimental Setup

- Experiments used 55 hour and 275 hour MGB3 challenge data for training and a 5.6 hour test set, dev17b.
- 63k word dictionary + trigram LM (tg) and confusion network decoding (cn).
- All models used cross-entropy training with extended HTK 3.5.
- Many ways of using $h_{t-n}$ in calculating $h_t$ (e.g. pooling, gating etc.). We focus on using $h_{t-n}$ as input to a HORNN layer.
- Standard HORNN model structures are ReLU: $h_t = \text{f}(Wx_t + Uh_{t-1} + Uh_{t-2} + b)$, Sigmoid: $h_t = \text{f}(Wx_t + Uh_{t-1} + Uh_{t-2} + b)$.
- Sigmoid HORNN requires to add extra $h_{t-2}$, since it suffers from vanishing gradients.
- HORNNs gave similar WERs, and used far less computation & space; without increasing WER.
- By going wider or deeper with similar #params, HORNNs outperformed LSTMs by ~4% relative WER reduction.

Conclusions

- Proposed to use high order (Markov) connections to address the RNN long-term vanishing gradient issue.
- Proposed two HORNN structures for sigmoid and ReLU.
- HORNNs yielded 4%-6% WER reductions over std. RNNs.
- Comparing to LSTMs/LSTMPs, HORNNs/HORNNPs
  - gave similar WERs, and used far less computation & space;
  - had 4% relative WER reduction with similar #params.