LIMITED-MEMORY BFGS OPTIMIZATION OF RECURRENT NEURAL NETWORK LANGUAGE MODELS FOR SPEECH RECOGNITION

Xunying Liu[1], Shansong Liu[1], Jinze Sha[2], Jianwei Yu[1], Zhiyuan Xu[2], Xie Chen[2] & Helen Meng[1]

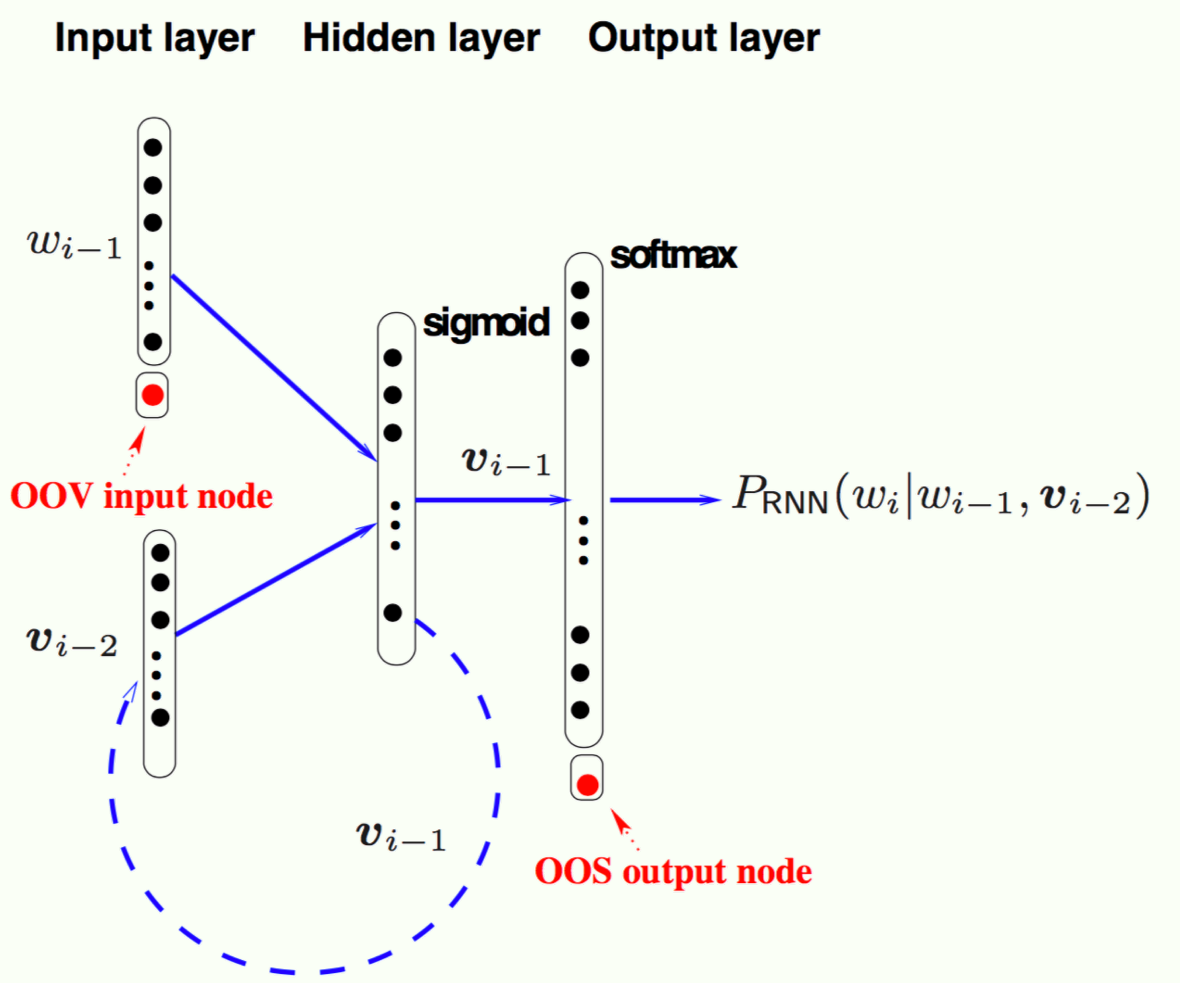
{xyliu, ssliu, jwyu, hmmeng}@se.cuhk.edu.hk, {js2294, zyx22, xc257}@cam.ac.uk

[1]The Chinese University of Hong Kong, [2]Cambridge University Engineering Department

Introduction

- **Problem statement and objectives**
  - Faster and more stable training for deep neural networks (DNNs)
  - Investigating 2nd order optimization techniques
  - Applied to recurrent neural network language model (RNNLM)
- **Existing RNNLM training algorithms**
  - Minimize the cross entropy (CE) using stochastic gradient descent (SGD)
  - SGD uses no higher order gradient information, models no correlation between parameters, poorly captures error cost function curvature
  - Quadratic approximation to error cost function using Newton methods
  - Storing and computing Hessian matrix and its inverse are problematic
  - Quasi-Newton methods, e.g. Hessian-free optimization applied to DNN acoustic modeling; iterative conjugate gradient (CG) method
  - CG search very expensive for large datasets in Hessian-free optimization
- **Our approach**
  - Limited-memory Broyden Fletcher Goldfarb Shannon (L-BFGS) based 2nd order optimization for RNNLM training
  - Efficiently approximates the product between inverse Hessian and gradient vector via a recursion over past gradients
  - Only require a few vectors representing finite number of past updates of the matrix-vector product, so it's memory efficient

Recurrent Neural Network LMs



- **RNNLM description**
  - Vector representation of complete word history $h_1^{i-1} = \langle w_{i-1}, \cdots, w_1 \rangle$
  - Sigmoid hidden layer activation
  - Shortlist output vocabulary plus out-of-shortlist (OOS) output node
  - Output probability linearly interpolated with n-gram LMs

$$P(w_i|h_1^{i-1}) = \lambda P_{NG}(w_i|h_1^{i-1}) + (1-\lambda) P_{RNN}(w_i|h_1^{i-1})$$

RNNLM Training Using SGD

- **Cross entropy training criterion**

$$J^{CE}(\boldsymbol{\theta}) = -\frac{1}{N_w}\sum_{i=1}^{N_w} \ln P_{RNN}(w_i|h_i)$$

  - where N_w is the total words of a given sequence
  - $P_{RNN}(w_i|h_i) = f_{softmax}(\boldsymbol{v}_{i-1}; \boldsymbol{\theta})$
- **SGD training procedure for RNNLM**
  - Parameter update: $\boldsymbol{\theta}[t+1] = \boldsymbol{\theta}[t] - \eta \left.\frac{\partial J^{CE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]}$

    $\boldsymbol{\theta}$ is the layer wise weight matrix, this is applied to all layers
  - Gradient stats for output layer weights: $\frac{\partial J^{CE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{1}{N_w}\sum_{i=1}^{N_w} v_i \xi_i^T$

    $\boldsymbol{\xi}_i$ is the error cost vector
    $\xi_{i,j} = \delta(w_j|h_i) - P_{RNN}(w_j|h_i)$
    $\delta(w_j|h_i) = 0$ or 1 (target prob.)
  - Back propagate, e.g. to recurrent layer:

$$\frac{\partial J^{CE}(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = -\frac{1}{N_w}\sum_{i=1}^{N_w} \boldsymbol{v}_{i-1}(\boldsymbol{\xi}_i \odot \boldsymbol{u}_i)^T$$

    $\boldsymbol{\zeta}$ is the recurrent layer weight matrix
    $\boldsymbol{u}_{i,j} = \boldsymbol{v}_{i,j}(1-\boldsymbol{v}_{i,j})$
    $\odot$ denotes elementwise multiplication
  - Back propagation through time (BPTT):

$$\frac{\partial J^{CE}(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = -\frac{1}{N_w}\sum_{i=1,\tau=1}^{N_w, N_\tau} \boldsymbol{v}_{i-\tau-1}(\boldsymbol{\xi}_{i-\tau} \odot \boldsymbol{u}_{i-\tau})^T$$

RNNLM Training Using L-BFGS

- **Newton methods**
  - Can model the correlation between model parameters using a quadratic approximation to the error cost function

$$J^{CE}(\boldsymbol{\theta}[t] + \Delta\boldsymbol{\theta}) \approx J^{CE}(\boldsymbol{\theta}[t]) + \Delta\boldsymbol{\theta}^T \left.\frac{\partial J^{CE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]} + \frac{1}{2}\Delta\boldsymbol{\theta}^T \boldsymbol{H}_t \, \Delta\boldsymbol{\theta}$$

  - Newton direction

$$\Delta\boldsymbol{\theta} = \boldsymbol{H}_t^{-1} \left.\frac{\partial J^{CE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]}$$

    where the Hessian matrix is computed as $\boldsymbol{H}_{t,i,j} = \left.\frac{\partial^2 J^{CE}(\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]}$
- **L-BFGS method for RNNLM training**

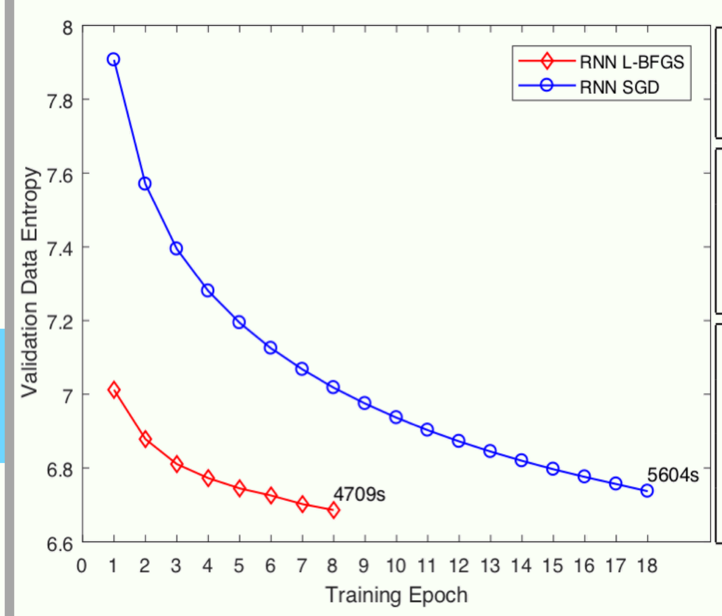
Algorithm 1 For RNNLM output layer weights $\boldsymbol{\theta}$, L-BFGS algorithm approximates inverse Hessian gradient matrix-vector product

1: $\boldsymbol{q}_t \leftarrow \left.\frac{\partial J^{\mathrm{CE}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]}$
2: **for** i = t-1, t-2, ..., t-m **do**
3: $\quad \boldsymbol{s}_i \leftarrow \boldsymbol{\theta}[i+1] - \boldsymbol{\theta}[i]$, $\boxed{\boldsymbol{y}_i \leftarrow \boldsymbol{q}_{i+1} - \boldsymbol{q}_i}$ (past gradient)
4: $\quad \rho_i \leftarrow \frac{1}{\boldsymbol{y}_i^\top \boldsymbol{s}_i}$, $\alpha_i \leftarrow \rho_i \boldsymbol{s}_i^\top \boldsymbol{q}_t$
5: **end for**
6: $\boldsymbol{B}_t^0 \leftarrow \frac{\boldsymbol{y}_{t-m}\boldsymbol{s}_{t-m}^\top}{\boldsymbol{y}_{t-m}^\top \boldsymbol{y}_{t-m}}$, $\boldsymbol{z} \leftarrow \boldsymbol{B}_t^0 \boldsymbol{q}_t$
7: **for** i = t-m, t-m+1, ..., t-1 **do**
8: $\quad \beta_i \leftarrow \rho_i \boldsymbol{y}_i^\top \boldsymbol{z}$, $\boldsymbol{z} \leftarrow \boldsymbol{z} + (\alpha_i - \beta_i)\boldsymbol{s}_i$
9: **end for**
10: $\boxed{\mathbf{H}_t^{-1} \left.\frac{\partial J^{\mathrm{CE}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[t]}} \leftarrow \boldsymbol{z}$ (parameter update direction)

- **Efficient GPU based training parallelization**
  - L-BFGS for RNNLM is implemented as an extension to CUED-RNNLM
  - Integrated into an efficient bunch mode GPU parallelization algorithm

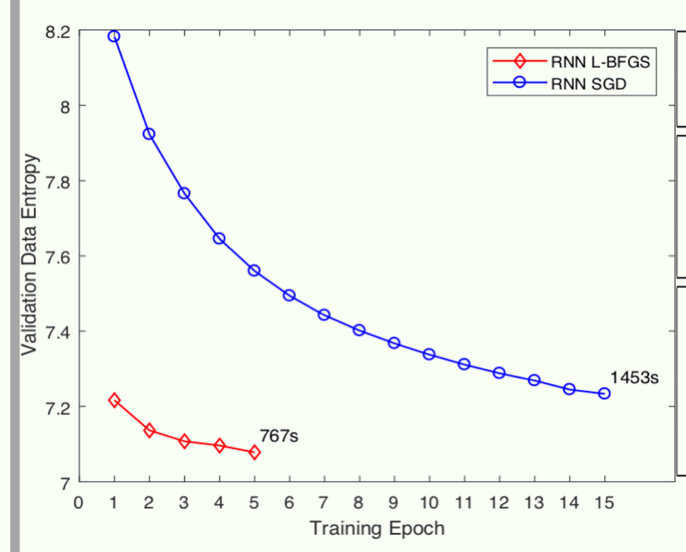
Experiments and Results

- **Experiment setup**
  - Datasets: the Switchboard English system (SWBD), 300 hours, 3.6M words of acoustic transcripts and a 30k words lexicon; Babel Cantonese system, 175 hours, 1.1M words of transcripts and a 25k vocabulary
  - Acoustic model: MPE trained stacked hybrid DNN-HMM by HTK toolkit
  - Target of acoustic DNN: SWBD, 12k tied states; Babel Cantonese, 6k tied states
  - RNNLM model structure and training: 512 hidden nodes and Sigmoid activation; SGD training with newbob scheduling plus momentum or L-BFGS method
  - Evaluation method: perplexity (PPL) and word error rate (WER)
  - GPU card: NVidia K40 GPUs; used to measure speed
- **Results on Switchboard**



LM	PPL	WER% swbd	WER% chm
rnn.SGD	104.3	13.9	26.1
rnn.LBFGS	100.8	13.2	25.9
rnn.SGD+rnn.LBFGS	**95.8**	**13.2**	**25.4**
4-gram	97.2	12.9	25.4
4-gram+rnn.SGD	87.0	12.6	24.8
4-gram+rnn.LBFGS	87.7	12.4	24.7
4-gram+rnn.SGD+rnn.LBFGS	**85.9**	**12.4**	**24.6**

  - Convergence: 19 epochs using 5604s for SGD; 9 epochs using 4709s for L-BFGS
  - 0.7% abs. WER reductions obtained by L-BFGS before interpolation with 4-gram
- **Results on Babel Cantonese**



LM	PPL	CER% devsub1	CER% devsub2
rnn.SGD	136.5	43.5	44.1
rnn.LBFGS	127.9	42.7	43.3
rnn.SGD+rnn.LBFGS	**119.7**	**42.6**	**43.2**
4-gram	113.7	42.1	42.7
4-gram+rnn.SGD	106.8	42.0	42.6
4-gram+rnn.LBFGS	106.2	41.8	42.4
4-gram+rnn.SGD+rnn.LBFGS	**104.8**	**41.8**	**42.3**

  - Convergence: 16 epochs using 1453s for SGD; 6 epochs using 767s for L-BFGS
  - 0.8% abs. WER reductions obtained by L-BFGS before interpolation with 4-gram
  - Observed on both tasks, the combination between SGD and L-BFGS is complementary since consistent improvements are obtained

Conclusion

- **L-BFGS optimization for RNNLM training & Future work**
  - Successfully applied to RNNLM training
  - Consistent improvements over SGD on multiple speech recognition tasks
  - Future research on L-BFGS training of advanced forms of NNs

This research was supported by MSRA grant no. 6904412 and Chinese University of Hong Kong (CUHK) grant no. 4055065