Assignment 4 Deep Learning
LSTM

Tian Wang
Center for Data Science
New York University
New York, NY 10003
tw991@nyu.edu

1 Question

Q1. See nngraph_handin.lua

Q2. $i = h_t^{l-1}, prev\_h = h_{t-1}^l, prev\_c = c_{t-1}^l$

Q3. Function create_network() returns a module containing all layers in a unrolled time step.

Q4. model.s stores the memory for each time step. model.ds stores the gradient. model.start_s is used to store the initialize memory unit in time 0 for each minibatch. Each time model starts to train or test on a new data, model.start_s needs to be set to be 0. Between minibatch, model.start_s for the next batch equals the final hidden state in the current batch.

Q5. When gradient is larger than a pre-set size, it will be shrunk to pre-set size.

Q6. Backprop from the end of a batch of sentences to the front and used gradient at the front to do stochastic gradient descent on this batch.

Q7. I set the extra output to be a zero tensor in the backward pass.

2 Description of the Architecture

2.1 Overall Structure

I followed the model architecture proposed in *Recurrent Neural Network Regularization*, Wojciech Zaremba et al. 2014.

Word-level LSTM I used a model containing 2 layers and unrolled for 20 steps. No dropout is applied. Each layer has 200 units. After 13 epochs, model achieved 141.418 perplexity on training set, 155.005 on validation set and 161.875 on testing set.

Character-level LSTM I used a model containing 2 layers and unrolled for 50 steps with dropout applied on non-recurrent input. Each layer has 1500 units. After 55 epochs, model achieved 240.372 on training set and 220.567 perplexity on validation set.

Other test I have also tested a model with 2 layers ,500 hidden units in each layer respectively and same training procedure with the baseline model, trying to see the performance improvement with bigger hidden unit size. After 20 epochs, I got a 53.751 training perplexity and 368.109 validation

perplexity. It's clear that model overfits the training data with more hidden units, causing the increase in validation perplexity. I think adding dropout to this model can help to reduce the gap between performance in training and validation.

3 Description of the Learning Techniques Applied

For word level LSTM, no dropout is applied and minibatch size is 300. For character level model, I applied the learning techniques used in *Recurrent Neural Network Regularization*. The hidden states are initialized to be 0, 65% dropout is applied on the non-recurrent connections, and the size of minibatch is 100.

4 Description of the Training Procedure

Character-level model is trained through 55 epochs. During the first 14 epochs, learning rate is 1. Afterwards, the learning rate is reduced by a factor of 1.15 each epoch. The norm of gradient is kept under 10. Perplexity is used as error metrics. The training of the network takes 1307 minutes on an NVIDIA K80 GPU. The training parameters are as following:

- max gradient norm: 10
- optimization method: mini-batch SGD
- rnn size: 1500
- initial weight: 0.04
- dropout rate: 0.65
- learning rate: 1
- batch size: 100
- learning rate decay(after 14 epochs): 1.15
- layer: 2

After 55 epochs, model achieved 240.372 perplexity on training set and 220.567 perplexity on validation set.

Word-level model has a similar training procedure, with total 14 epochs and 2 learning rate reduce factor after 4 epochs. It achieved 141.418 perplexity on training set, 155.005 on validation set and 161.875 on testing set.

References

Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." arXiv preprint arXiv:1409.2329 (2014).*arXiv preprint arXiv:1409.2329*, 2014.