DS-GA-1008: Deep Learning Spring 2015 RNN on Word Level and Character Level Prediction

Yen-Cheng Liu
ycl391@nyu.edu

Abstract

The goal of this assigiment is to understand the structure of RNN in natural language processing, Including the lstm model and details of how forward propagation and backward propagation work in RNN network. At beigining. we try to dipict the network describing in th code and add the preictions into ouput of the model. Second, we add the query_sentense() function which is modified from run_test() so that we can use it to execute simple test for a given string and specified number of word length that we want o predict. Finally, we modify the whole code from world-level to character level.

1 Answers

Q1 : please seee Q1.lua
Q2 : i = h_t^{l-1},prev_c = c_{t-1} , prev_h = h_{t-1}^l
Q3:It creates the rolled 2-layers newtwork, including 2 lstm components(default setting) , after calling model.rnns = g_cloneManyTimes(core_network, params.seq_length) in setup(), now model.rnns becomes unrolled.
Q4 : model.s record the state after executing each step in fp and bp. model.ds is the gradient of the state. model.start_s is the initial state. those are reset to zero at the beigining of training and any testing.
Q5 : L2 norm
Q6 : Stochastic gradient descent(SGD) and clipping gradient.
Q7 : Just simply add pred variable into outputs of nn.gModule()(). After that we want to modify backpro. In order to do not affect the backpro process by the additional output, we just create a tensor call dummy_depred = transfer_data(torch.zeros(params.batch_size,params.vocab_size)) and feed it into backprob. Setting dummy_depred to all zero means no matter what pred is, it gradient values are always zero.This does not affect backprob.

2 Architecture

The whole structure is the same as baseline model. However, we test different settings including layers, Droupout, RNN size and grad_norm. In word level, setting params = batch_size=20, seq_length=20, layers=3, decay=2, rnn_size=200, dropout=0.3, init_weight=0.1, lr=1, vocab_size=10000, max_epoch=4, max_max_epoch=13, max_grad_norm=4 produces validation set perplexity 107.664 and test set perplexity 104.990. In character levle testing, we use params = batch_size=20, seq_length=20, layers=3, decay=2, rnn_size=100, dropout=0.3, init_weight=0.1, lr=1, vocab_size=50, max_epoch=4, max_max_epoch=13, max_grad_norm=4

3 Discussion

- Learning rate : here we set default value.

- Gradient clipping: This can prevent over-descend. In this assignment, we find that this value can be larger in character level(up to 10) and smalled in word level. Here we choose 5 or 4.
- Dropout: In order to increase the power of network and reduce over fitting, we test different setting and finally choose 0.3 to 0.5(general setting)
- Layers :Layer number can improve the correctness of this network, I've tested layer number from 2 to 5. However larger layer number can improve the network but also take more time. Finally I use set layer number to 3. Because I do not have much time to find best setting.
- RNN Size: This represent the dimension of feature vector of each word or character. Here we just use initial random setting. However, maybe we can use glove word feature vector to represent it and remove the backprob for LookUpTable. Maybe this will reduce the training time and improve the accuracy.

4 Summary

To understand the network was a tough work. However after realizing it, it is a powerful Architecture. Since this is a two weeks homework, we don't have to much time to tune best settings. larger layer and RNN size may take one or two days to train a network. Seems like we should spend more time to study RNN.

References

[1] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), 12.*

[2] Zhang, X., & LeCun, Y. (2015). Text Understanding from Scratch. *arXiv preprint arXiv:1502.01710.*

[3] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research, 12, 2493-2537.*