Report for Spring 2019 PRML Assignment # 3

Date: May 20, 2019

Abstract: The purpose of this assignment was to gain experience with implementing RNN especially LSTM. We will learn how to differentiate LSTM. Furthermore, we will use LSTM to generate Tang poetries.

1 Part 1: Differentiate LSTM

1.1 Differentiate one step of LSTM with respect to h_t

$$\frac{\partial h_t}{\partial o_t} = \tanh(C_t)$$

$$\frac{\partial h_t}{\partial o_t} = o_t * \frac{\partial \tanh(C_t)}{\partial C_t} = o_t * (1 - \tanh^2(C_t))$$

$$\frac{\partial h_t}{\partial f_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial f_t} = o_t * (1 - \tanh^2(C_t)) * C_{t-1}$$

$$\frac{\partial h_t}{\partial i_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial i_t} = o_t * (1 - \tanh^2(C_t)) * \overline{C}_t$$

$$\frac{\partial h_t}{\partial \overline{C}_t} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial \overline{C}_t} = o_t * (1 - \tanh^2(C_t)) * i_t$$

$$\frac{\partial h_t}{\partial C_{t-1}} = \frac{\partial h_t}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} = o_t * (1 - \tanh^2(C_t)) * f_t$$

$$\frac{\partial h_t}{\partial W_o} = \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial W_o} = \tanh(C_t) * o_t * (1 - o_t) * z$$

$$\frac{\partial h_t}{\partial b_o} = \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial b_o} = \tanh(C_t) * o_t * (1 - o_t)$$

$$\frac{\partial h_t}{\partial W_i} = \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial W_i} = o_t * (1 - \tanh^2(C_t)) * \overline{C}_t * i_t * (1 - i_t) * z$$

$$\frac{\partial h_t}{\partial b_i} = \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial b_i} = o_t * (1 - \tanh^2(C_t)) * \overline{C}_t * i_t * (1 - i_t)$$

$$\frac{\partial h_t}{\partial W_C} = \frac{\partial h_t}{\partial \overline{C}_t} * \frac{\partial \overline{C}_t}{\partial W_C} = o_t * (1 - \tanh^2(C_t)) * i_t * (1 - \overline{C}^2) * z$$

$$\frac{\partial h_t}{\partial b_C} = \frac{\partial h_t}{\partial \overline{C}_t} * \frac{\partial \overline{C}_t}{\partial b_C} = o_t * (1 - \tanh^2(C_t)) * i_t * (1 - \overline{C}^2)$$

$$\frac{\partial h_t}{\partial W_f} = \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial W_f} = o_t * (1 - \tanh^2(C_t)) * C_{t-1} * f_t * (1 - f_t) * z$$

$$\frac{\partial h_t}{\partial b_f} = \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial b_f} = o_t * (1 - \tanh^2(C_t)) * C_{t-1} * f_t * (1 - f_t)$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial h_{t-1}} + \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial h_{t-1}} + \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial h_{t-1}} + \frac{\partial h_t}{\partial \overline{C}_t} * \frac{\partial \overline{C}_t}{\partial h_{t-1}} = o_t * (1 - \tanh^2(C_t)) *$$

$$(C_{t-1} * f_t * (1 - f_t) * W_h f + \overline{C}_t * i_t * (1 - i_t) * W_h i + i_t * \left(1 - \overline{C}^2\right) * W_h C) + \tanh(C_t) *$$

$$o_t * (1 - o_t) * W_h o$$

$$\frac{\partial h_t}{\partial x_t} = \frac{\partial h_t}{\partial f_t} * \frac{\partial f_t}{\partial x_t} + \frac{\partial h_t}{\partial i_t} * \frac{\partial i_t}{\partial x_t} + \frac{\partial h_t}{\partial o_t} * \frac{\partial o_t}{\partial x_t} + \frac{\partial h_t}{\partial \overline{C}_t} * \frac{\partial \overline{C}_t}{\partial x_t} = o_t * (1 - \tanh^2(C_t)) * (C_{t-1} * f_t *$$

$$(1 - f_t) + \overline{C}_t * i_t * (1 - i_t) + i_t * \left(1 - \overline{C}^2\right)) + \tanh(C_t) * o_t * (1 - o_t)$$

1.2 Differentiate Through Time

For time = k, we can use the following equation

$$\frac{\partial h_t}{\partial h_k} = \frac{\partial h_t}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_{t-2}} * \frac{\partial h_{t-2}}{\partial h_{t-3}} * \frac{\partial h_{t-3}}{\partial h_{t-4}} \dots \dots = \prod_{i=k}^{t-1} \frac{\partial h_{i+1}}{\partial h_i}$$

2. Part 2: Autograd Training of LSTM

2.1 Initialization

If the weights are set to 0, it will cause the all the derivative with respect to the loss function for W to be equal. Which will make the hidden units symmetric and continues for all the iterations. One way to initialize the weights is using random, however, this could lead to gradients that are too big or too small. A better way is to use gradient clipping, which sets a threshold value.

2.2 Generating Tang poem.

To make generating poems easier, I used the 全唐诗 from the start. 16384 poems are picked as training set, and 2730 poems as development set. For perplexity I used $\sum \log(1/p(s_i|s_{i-1}, s_{i-2}, \dots)$.

日：日衣华楼春， 白首郭乱背。胡一万卷村，岁间马陆镇。

红：红日白云方误差，眼前水川可无视。悠见哀老不见翁，清风树种白头吟。

山：山人就已先，高书马一鸣。人人不得有，无穷人老生。

夜：夜市雨纷纷，穷奇烦心事。病衰帝风雨，旧时代天灯。

湖：湖锋白鹭蜻蜓舞，故园宁辞新市路。自外云鹏颖浇怀，国我昌吹灰烬开。

海：海水化神飘，山头方禄来。可是异乡人，不知何处来。

月：月酒黑夜无尽赖，我把清水不知几。谁能问世相对君，不知几时是几时。

2.3 Optimization

For optimization I tried SGD with momentum and Adam. The SGD with momentum has the benefit of being fast, however, it is quite unstable which needs a lot tuning for the learning rate. The Adam is more stable, but the speed is slower.

3 Conclusion

Overall, the assignment hard since I did not know how to use PyTorch and FastNLP. I tried my best to do as much as I could to finish the assignment. In the end, I generated the poems as the best I could. So I consider this assignment a success.

References

1. “Assignment 3” Pattern Recognition and Machine Learning, Fudan University, Spring 2019,

< https://zfhu.ac.cn/PRML-Spring19-Fudan/assignment-3/index.html#dfref-footnote-6>.