Weekly Report(May 21 - May 27)

Liu Junnan

Abstract

This week I finished cs231n course and started to do assignment3.

1 Work Done

1.1 RNN and LSTM

RNN Unlike feedforward neural networks, recurrent neural networks take as their input not just the current input example they see, but also what they have perceived previously in time. The decision a recurrent net reached at time step $t-1$ affects the decision it will reach one moment later at time step t. So RNNs have two sources of input, the present and the recent past. The sequential information is preserved in the recurrent network's hidden state, which manages to span many time steps as it cascades forward to affect the processing of each new example. One way to think about RNNs is this: they are a way to share weights over time. Therefore RNNs are good at processing sequences like image caption and machine translation.

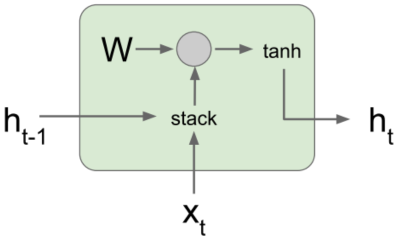

Figure 1: RNN unit

The forward pass of a RNN block can be mathematically described as:

$$h_t = f_W(h_{t-1}, x_t)$$

where h_t is the current hidden state at time step t; f_W the function – either sigmoid or tanh – that squashes the sum of the weight input and hidden state, making gradients workable for backpropagation; h_{t_1} the previous hidden state; x_t the input at the same time step.

Recurrent networks rely on an extension of backpropagation called backpropagation through time, or BPTT. Time, in this case, is simply expressed by a well-defined, ordered series of calculations linking one time step to the next, which is all backpropagation needs to work.

However, RNNs suffer a lot from vanishing/exploding gradient problems. Intuitively, backpropagation from h_t to h_{t-1} multiplies by W_h, so computing gradient of h_0 involves many factors of W_h. If the largest singular value of W_h is greater than 1, then exploding gradients would occur; on the contrary if the largest singular value of W_h if less than 1, then vanishing gradients would happen.

Exploding gradients can be solved relatively easily, because they can be truncated or squashed. Vanishing gradients can become too small for computers to work with or for networks to learn – a harder problem to solve.

LSTM Long short-term memory was proposed to solve vanishing gradient problem mentioned above. LSTMs design complicated cells that help preserve the error that can be backpropagated through time and layers.

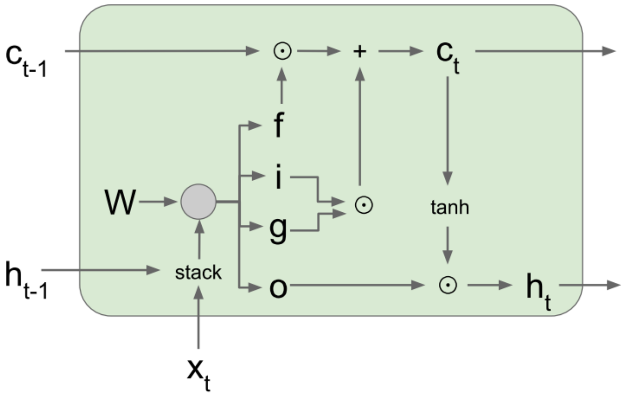

Figure 2: LSTM cell

The forward pass of a LSTM cell is defined as follows:

$$\begin{aligned}\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t_1} \\ x_t \end{pmatrix} \\ c_t &= f \odot c_{t_1} + i \odot g \\ h_t &= o \odot \tanh(c_t)\end{aligned} \tag{1}$$

where $\odot$ is element-wise multiplication.

As described in equation(1)

2 Plans