Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning – Supplementary Material

Pingbo Pan[§] Zhongwen Xu[†] Yi Yang[†] Fei Wu[§] Yueting Zhuang[§]
[§]Zhejiang University [†]University of Technology Sydney
{lighnt001,zhongwen.s.xu}@gmail.com yi.yang@uts.edu.au
{wufei,yzhuang}@cs.zju.edu.cn

In this supplementary material, we perform experiments on the Microsoft Video Description Corpus (MSVD) dataset to study the choice of parameters in our proposed Hierarchical Recurrent Neural Encoder (HRNE), then compare the speed of multi-layer Long Short-Term Memory (LSTM) and HRNE.

1. Parameter Study

To choose proper parameters for our method, we perform experiments with different parameter settings. We first study how to choose the number of hidden units and report the results in Table 1. For simplification, we set the number of hidden units in LSTMs from the encoder and the decoder as the same. The result shows that the performance of our method can be improved by increasing the number of hidden units. Nevertheless, if the number of hidden units is larger than 1,024, our method is more likely to suffer from the over-fitting problem.

Number of hidden units	256	512	1,024	2,048
METEOR	31.9	32.3	33.1	32.7

Table 1: Experiment results on MSVD dataset with different numbers of hidden units

We also study the choice of temporal filter size, *i.e.*, the length of the LSTM chain in the first layer, in Table 2. The result shows that our method is insensitive and robust to the choice of filter size.

Filter size	4	8	16
METEOR	32.1	33.1	33.0

Table 2: Experiment results on MSVD dataset with different filter sizes.

num of layers	Stacked LSTM	HRNE	HRNE w/ att
2	4.2	2.5	3.0
4	6.3	2.7	3.3

Table 3: A comparison of the running time (in second) of different methods. It shows the training time of a video batch with size 128 on the MSVD dataset. "HRNE w/ att" in the last column denotes HRNE with attention.

2. Speed Comparison

We compare the running time of multi-layered LSTM and our proposed HRNE. All of the running time reported is on a single video batch of size 128 in training phase (including the backpropagation time), which is averaged over 10 mini-batches. Experiments are performed on an NVIDIA GTX Titan X. Table 3 indicates our method can reduce computation operations and adding attention mechanism slightly increases calculation time. Since our method adopts pyramid-shaped structure, we notice doubling LSTM layers from 2 to 4 mildly increases training time from 2.5 to 2.7, while it makes huge computation overhead for the multi-layered LSTM.