CPSC 8430 -HW2

Fangjian Li

March 2021

The codes are available at: https://github.com/FangjianLi/Deep-learning-HW2

1 Sequence-to-sequence (seq2seq) model

1.1 Concepts of seq2seq model

In this homework, a typical sequence to sequence model Venugopalan et al. (2015) is first adopted to do the video-captioning task.

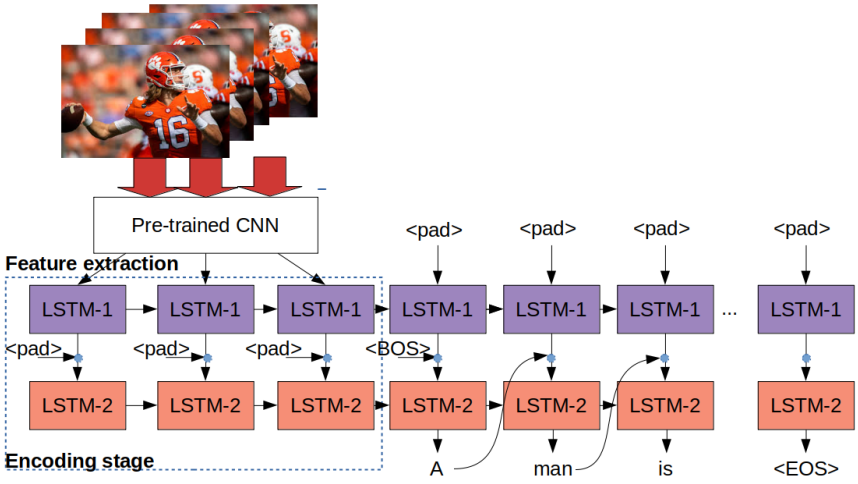

Figure 1: Block diagram of seq2seq

The video captioning generation process is illustrated in Fig. 1. The features of the videos are extracted based on the pre-trained convectional neural networks (CNN). There are two LSTMs in this model. In the encoding stage, the features in different time steps are injected into the first LSTM sequentially. There is no need for the second LSTM to produce the outputs in this stage. The encoding stage finishes after all the features have been injected. In the decoding stage, the padding

(zero vector) is injected into the first LSTM. The second LSTM will generate the caption of the videos step by step. The sequence of the caption starts with the token $< BOS >$, i.e., the beginning of the sentence, ends with the token $< EOS >$, i.e., the end of the sentence.

1.2 Build the model

The sequence-to-sequence model is built mainly based on the ***Tensorflow-v1*** package. The model is composed of three components, i.e., the data processing, the model training, and the model testing.

For the data processing, the ***pandas*** package is used to read the jason data files and build the data table contains "caption" and "video_feature_paths". For training efficiency and consistence purpose, the shortest caption of each video has been chosen as the training label. In addition, the words-to-indexes dictionary and indexes-to-words dictionary are built and pickled accordingly. Corresponding to the coding repository, the above process are realized in ***utils_data_processing_1.py***.

In the modeling training scripts, the ***tf.nn.rnn_cell.BasicLSTMCell*** function is mainly used to build the LSTM model. In the decoding stage, to feed the ground-truth data into the next time step, ***tf.nn.embedding_lookup*** function is used to lookup the desired output corresponding to the ground-truth label. The cross entropy losses in the decoding stage are summed up as the training loss. In addition, the ***caption_mask*** variable is introduced and applied to the entropy loss in case the caption length if smaller than the default decoding stage length. The batch sampling process is quite straightforward. The only tricky steps might be ***keras.preprocessing.sequence*** package is used to make the dimension of the caption consistent. Once the sampling process is done, the Adam optimizer is used to minimize the training loss defined above. The above process are realized in the ***model_seq2seq.py*** (modeling and training) and ***utils_data_sampling_1.py*** (sampling).

The testing process are written based on the structure of ***tensorflow.saver()***, i.e., the graph is built in the beginning and the parameters are restored then. Since the ground-truth captions are no longer available, the decoder needs to feeds its output as the next time step's input.

1.3 Training results

The training configurations are shown in the Table. 1. It has been found that, as the shortest labels are used to train the model, 60 training epochs are good enough to have a good training result. By using the Clemson Palemtto computing resource (16xCPU, 64gb RAM, 2GPU), the training time is around 36 mins.

Table 1: Seq2seq model training configuration

Optimizer	Adam	**Learning rate**	1e-4
# of epochs	60	**Loss function**	cross entropy
Batch size	128	**Feature size**	4096x80
Encoder hidden state size	512	**Decoder hidden state size**	512
Encoder step length	80	**Decoder step length**	20

The loss versus the training epochs have been provided in Fig. 2. As can be seen, the loss reduction is fastest in the beginning and then become slower as the training involved. Based on the provided dataset, the Bleu score is **0.6258**, which is not very appealing. Extra training dataset might be able to improve the performance of the seq2seq model. The codes are in the folder: ***hw2/Others/Seq2seq/***.

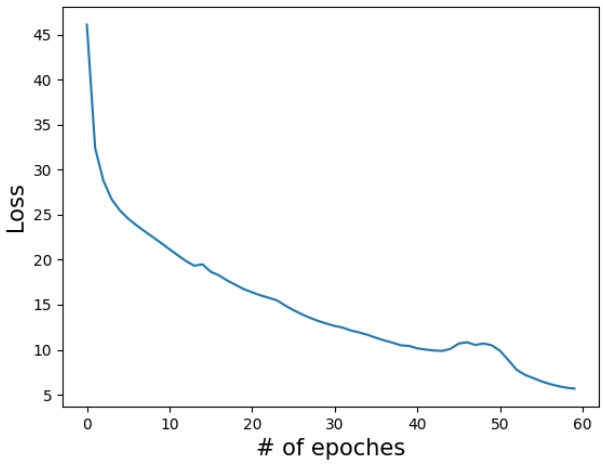

Figure 2: Loss versus training epochs

2 Add more modifications

In this section, we will add modifications on top of the seq2seq model to improve its performance.

2.1 The attention

When finishing the encoding, the encoder information is only feed in the decoder once in the beginning of the decoding stage. This is not how human translate a sentence.

As a result, the Bahdanau Attention algorithm Bahdanau et al. (2014) is used here. We assume that decoder RNN's state at t^{th} time step as s_t. In the encoding stage, the RNN's stage at the j^{th} time step as h_j. The coefficients of the encoder state h_j at time step t can be calculated as follows:

$$e_{jt} = V_{att}^T tanh(U_{att} s_{t-1} + W_{att} h_j) \tag{1}$$

where $V \in R^d$, $U \in R^{dxd}$, $W \in R^{dxd}$, which can be learned in the training process.

$$\alpha_{jt} = \frac{exp(e_{jt})}{\sum_{j=1}^{N} exp(e_{jt})} \tag{2}$$

After that, $c_t = \sum_{j=1}^{T} \alpha_{jt} h_j$ is injected to the RNN module at time step t as an extra input. The diagram is illustrated in the Fig. 3.

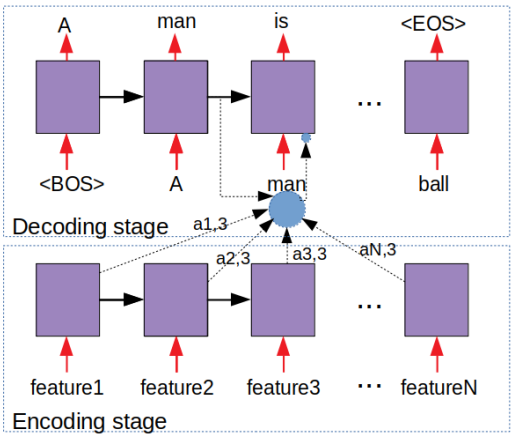

Figure 3: The attention mechanism

The modifications are directly fulfilled by introducing the extra padding for the LSTM2 during the encoding stage. During the decoding stage the c_t is calculated based on the batch multiplication in Tensorflow. To improve the computation feasibility in the local laptop, the decoder step length has been reduced to 18. The number of epochs have been extended to 70. The rest of training configurations/hyperparameters are same as Table 1. The Bleu score for this model

(seq2seq+attention) is **0.6393**, which improves compared to the conventional seq2seq model in section 1. The codes are in the folder: ***hw2/Others/Seq2seq_attention/***.

2.2 Schedule sampling

In the schedule sampling mechanism Bengio et al. (2015), the ground-truth and the last time step's output are feed to the next RNN at odds. In the conventional algorithm, the training process is guided by the ground-truth. This might cause the problem in the testing/prediction process when the ground truth information is no longer available. As a result, in schedule sampling mechanism, the last time step's output is involved more often as the training process goes on. The Tensorflow ***tf.cond()*** function is used to fulfill the switch between the ground truth input and the last time step's output. The training configurations/hyperparameters are same as Table 1. The Bleu score for this model (seq2seq+schedule sampling) is **0.6673**, which is considered as a good improvement. The codes are in the folder: ***hw2/Others/Seq2seq_ss/***.

2.3 Attention+schedule sampling

Finally, we combined the above methods together and build the seq2seq+attention+schedule sampling model. The configuration/hyperparameters are same as section 2.1, which is also detailed in Table 2.

Table 2: Seq2seq+attention+schedule sampling model training configuration

Optimizer	Adam	**Learning rate**	1e-4
# of epochs	70	**Loss function**	cross entropy
Batch size	128	**Feature size**	4096x80
Encoder hidden state size	512	**Decoder hidden state size**	512
Encoder step length	80	**Decoder step length**	18

The Bleu score is **0.6764**, which is a very good improvement comparing to the conventional seq2seq model. The codes are in the folder: ***hw2/hw2_1/***. This is also the final version of the submission to this homework.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014), "Neural machine translation by jointly learning to align and translate." *arXiv preprint*

arXiv:1409.0473.

Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer (2015), "Scheduled sampling for sequence prediction with recurrent neural networks." *arXiv preprint arXiv:1506.03099*.

Venugopalan, Subhashini, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko (2015), "Sequence to sequence-video to text." In *Proceedings of the IEEE international conference on computer vision*, 4534–4542.