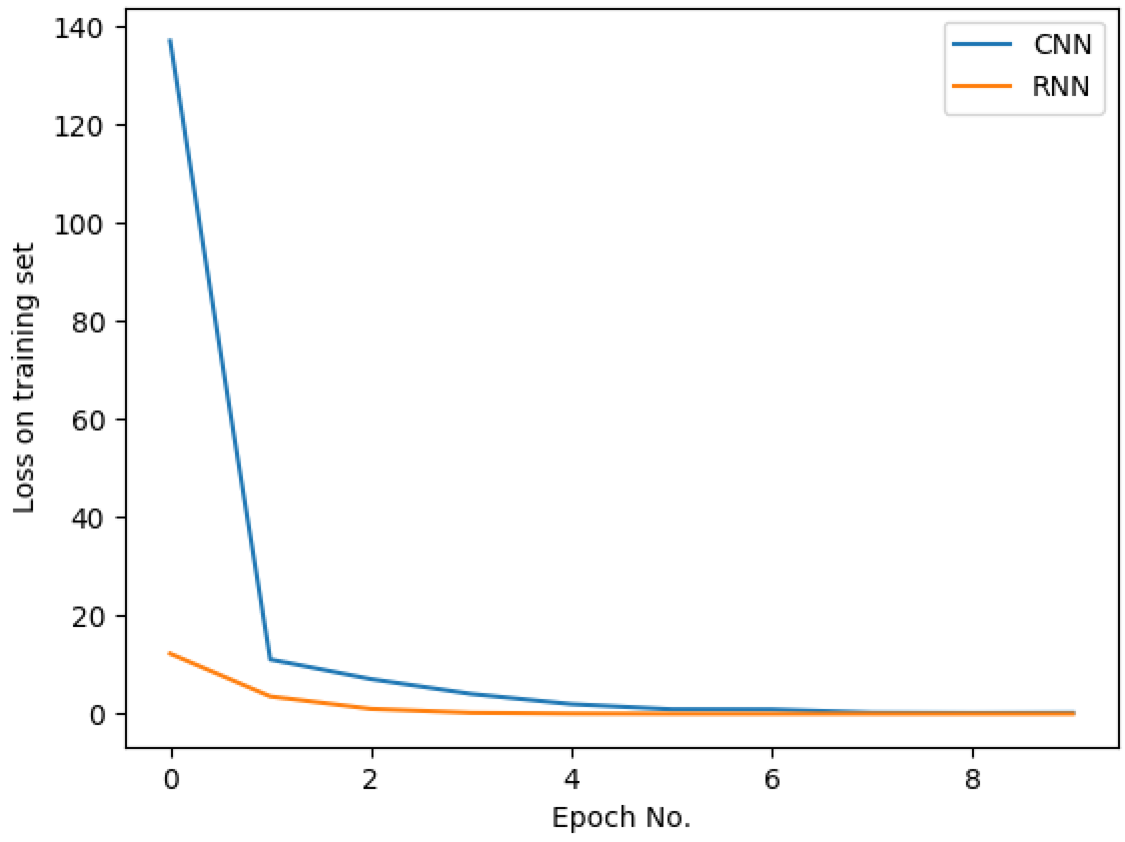
RNN and CNN

Part I Implementation of RNN and CNN

The loss converges rapidly in CNN and RNN as the pictures shown below indicates. However, they can easily overfits the training set thus having poor performance on development set. After 10 epochs, my accuracy on training set reaches to 0.998 for CNN and 1.0 for RNN, while 0.704 and 0.727 on test set.

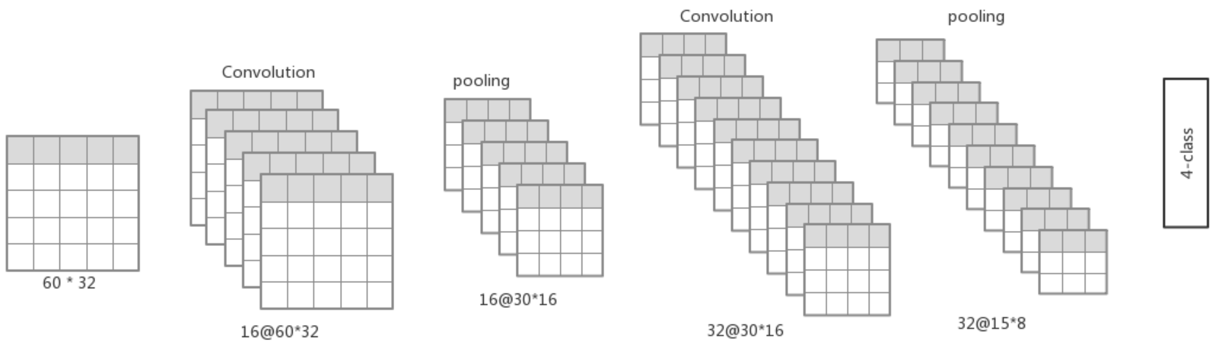


Comparasion of CNN and RNN

Details on implementation

My embedding size is 32 and batch size is 16.

The CNN is divided into two parts: a block of convolutional layers and fully connected layer. The convolutional layer is composed of a 5*5 convolution kernel and 2*2 maximum pooling. To make the output and input have the same height and width we set the padding to 2 and stride to 1. The implementation is shown below.



And for my RNN I set the hidden size to 5 to ease the model.

Part II Thoughts of FastNLP

Although FastNLP may help us better deal with our lab, there are still some problems.

- First, the document need to be more detailed and offers easier access. Examples to illustrate the API need to be detached from the source code.
- For API Instance, there are many data types not supported, so users have to deal with these minor details by themselves. However, the feedback sometimes is too weak to find where the bug is.
- For API Batch, maybe another keyword should be offered to users to determine whether to drop out the last batch which may not amounts to the required batch number. And it may be a little confusing that Batch API return two examples, one consisting of keyword "input" and one consisting of keyword "target", so related document should be more explicit on these minor details.

References

Morvan zhou's excellent demo of CNN

Aston Zhan, et al. (Still under revision) Dive into Deep Learning.