计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目：Fun with RNN		学号：201600181058
日期：2019-4-25	班级：16 人工智能班	姓名： 张多
Email：976539567@qq.com		

实验目的：

Configure the cloud server and test it.

实验软件和硬件环境：

Python 3.6

ThinkPad X1Carbon 8G+256G

Huawei cloud server

Experiment Principles and methods:

RNN is a kind of special neural networks that can memorize the state of a sequence. The input and the output of the network are mixed with the previous state information and the amount of the memory is determined by will. The main approach of memorizing a sequence is modifying the generation of the hidden layer. Here are the details:

1. RNN model:

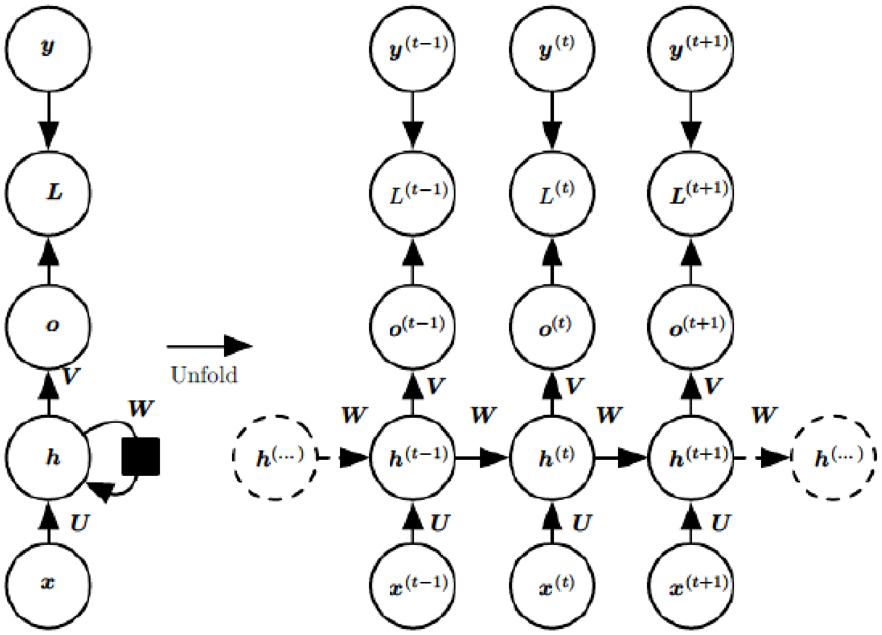


Figure1. The illumination of RNN

The left part of the image above is the RNN model without unfolding with respect to time. If it’s unfolded with respect to its time series, the architecture is going to be like the right part.

2. Notations:

- t represents the index of the time sequence.
- x, o are the input and the output of the network
- h is the hidden layer which represents the hidden state of the layer

- y is the real output that is respond to the input x
- L is the value of the loss function at time t
- U, W, V are the parameters of the linear relation, which are shared in the whole network and perfectly demonstrate the spirit of recurrent.

3. Forward pass:

Initially, h should be considered because its significance to the whole network, and $\mathrm{h}^{(t)}$ is derived by $\mathrm{x}^{(t)}$, $h^{(t-1)}$:

$$\mathrm{h}^{(\mathrm{t})} = \sigma\left(z^{(t)}\right) = \sigma\left(Ux^{(t)} + \; Wh^{(t-1)} + b\right) \tag{1}$$

Where σ is the activation function, which usually is tanh and b is the linear bias.

The output of the model is:

$$\mathrm{o}^{(\mathrm{t})} = Vh^{(t)} + c \tag{2}$$

The prediction of the model is:

$$\hat{y}^{(t)} = \sigma\left(o^{(t)}\right) \tag{3}$$

Generally, as a result of the purpose of RNN, which is to implement the classification for series, the activation function are usually softmax and the loss function are the log-crossentropy.

Back Propagation

As for RNN, because the losses are distributed on every index of the sequence, the ultimatum Loss is:

$$L = \sum_{\mathrm{t}=1}^{\tau} L^{(t)} \tag{4}$$

The gradient of V, c is easier to calculate:

$$\frac{\partial L}{\partial c} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial c} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial o^{(t)}} \frac{\partial o^{(t)}}{\partial c} = \sum_{t=1}^{\tau} \left(\hat{y}^{(\mathrm{t})} - \mathrm{y}^{(\mathrm{t})}\right) \tag{5}$$

Where the detailed simplification process is demonstrated in the report of ex1.

$$\frac{\partial L}{\partial V} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial V} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial o^{(t)}} \frac{\partial o^{(t)}}{\partial V} = \sum_{t=1}^{\tau} \left(\hat{y}^{(\mathrm{t})} - \mathrm{y}^{(\mathrm{t})}\right)\left(h^{(t)}\right)^{T} \tag{6}$$

To simplify the induction process, assume the gradient of the hidden state at index t is:

$$\delta^{(\mathrm{t})} = \frac{\partial L}{\partial h^{(t)}} \tag{7}$$

With $\delta^{(\mathrm{t})}$, the gradients of W, U, b are easier to derive:

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial W} = \sum_{t=1}^{\tau} diag\left(1 - \left(h^{(t)}\right)^{2}\right) \delta^{(t)} \left(h^{(t-1)}\right)^{T} \tag{8}$$

$$\frac{\partial L}{\partial b} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial b} = \sum_{t=1}^{\tau} diag\left(1 - \left(h^{(t)}\right)^{2}\right) \delta^{(t)} \tag{9}$$

$$\frac{\partial L}{\partial U} = \sum_{t=1}^{\tau} \frac{\partial L^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial U} = \sum_{t=1}^{\tau} diag\left(1 - \left(h^{(t)}\right)^2\right) \delta^{(t)} \left(x^{(t)}\right)^T \tag{10}$$

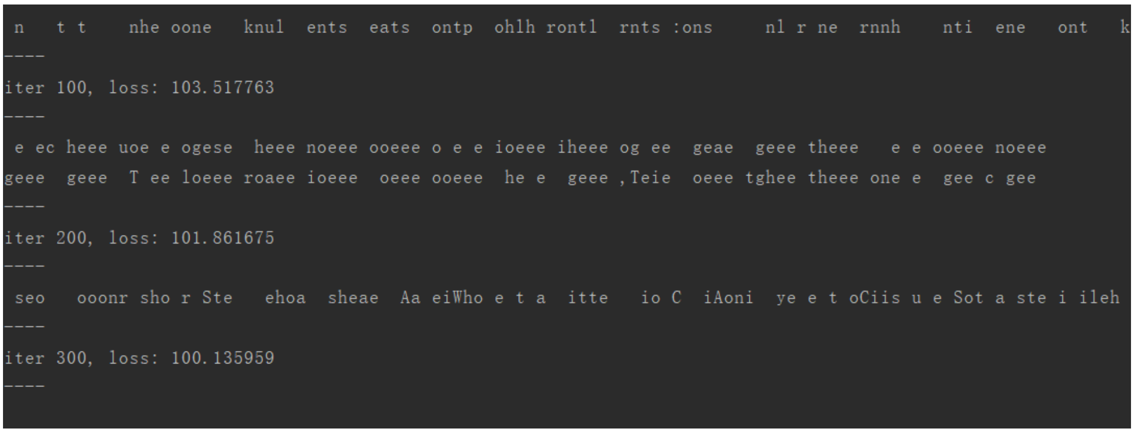
实验步骤：（不要求罗列完整源代码）

1. Complete the code according to the equations above.
2. Train the RNN
3. Change the temperature and test the performance.
4. Using the given parameters to generate a string.
5. Analyze the result.

结论分析与体会：

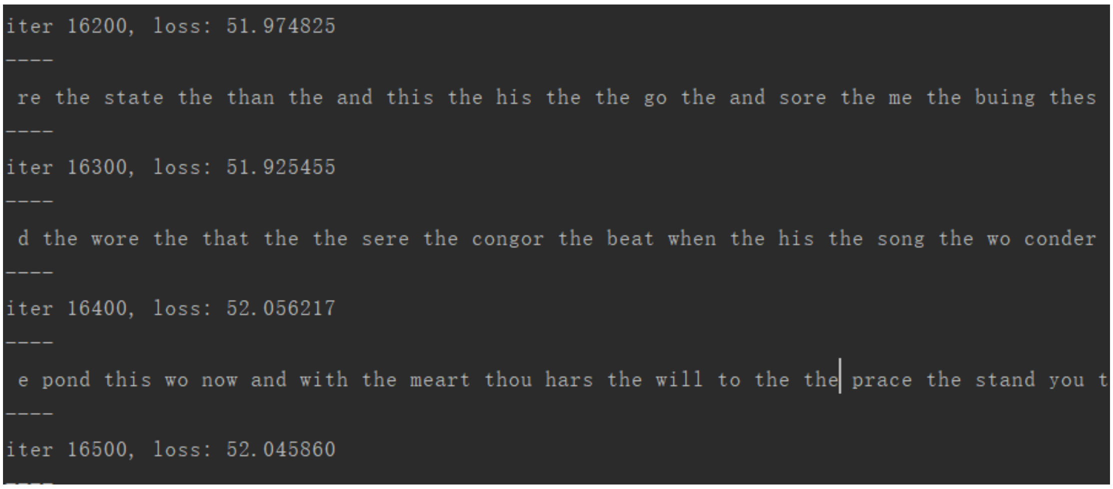
1. When training the RNN, the loss decreased to about 50 quickly and after that the loss descent speed started a plunge.

   At the beginning the letter combinations are quite random



```
 n   t t    nhe oone   knul  ents  eats  ontp  ohlh rontl  rnts :ons     nl r ne  rnnh    nti  ene   ont   k
----
iter 100, loss: 103.517763
----
 e ec heee uoe e ogese  heee noeee ooeee o e e ioeee iheee og ee  geae  geee theee   e e ooeee noeee
geee  geee  T ee loeee roaee ioeee  oeee ooeee  he e  geee ,Teie  oeee tghee theee one e  gee c gee
----
iter 200, loss: 101.861675
----
 seo   ooonr sho r Ste   ehoa  sheae  Aa eiWho e t a  itte   io C  iAoni  ye e t oCiis u e Sot a ste i ileh
----
iter 300, loss: 100.135959
----
```

Figure2. The string generation at the beginning of the training



```
iter 16200, loss: 51.974825
----
 re the state the than the and this the his the the go the and sore the me the buing thes
----
iter 16300, loss: 51.925455
----
 d the wore the that the the sere the congor the beat when the his the song the wo conder
----
iter 16400, loss: 52.056217
----
 e pond this wo now and with the meart thou hars the will to the the prace the stand you t
----
iter 16500, loss: 52.045860
----
```

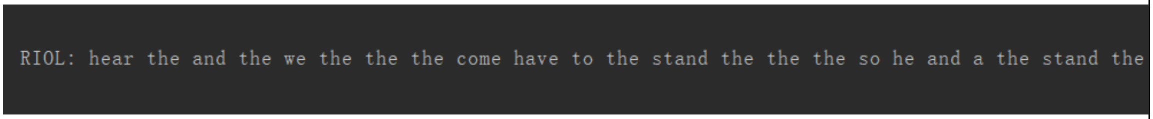
Figure3. The string generated when the loss is down

   Now the words generated are obviously clearer and more correct. The RNN is fundamentally established.

2. The temperature mainly affects the non-normalized output probability. Set $\tau = 1$ so that the probability is unaffected. However, if $\tau > 1$, because of the position of the temperature which is on the dominator, the probability is greatly decreased and the

difference between the right probability and the wrong one is shrank or even obliterated. This is bad because the memory is juggled and distorted and finally this jeopardizes the performance of the RNN. On the contrast, if $\tau < 1$, so the difference among probabilities are greatly augmented and the right result will be prominent with a better performance.

3. String generation



```
RIOL: hear the and the we the the the come have to the stand the the the so he and a the stand the
```

Figure4. The string generated by the samples.txt

The words are integral.

4. I really don't know the reason of the phenomenon. However, I think the matrix h of the colon or other punctuations has a strong respond of space or start a new line. The h are maybe mostly zeros and cause the same result every time.

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题：

I can barely understand RNN and its derivative. I searched so many references and finally I understand it.