Report: RNN

1. Introduction to recurrent neural networks

The feedforward neural network assumes that each input is independent, that is to say, the output of each network depends on the current input only. However, in many tasks, input at different times can affect each other, such as video, voice, text and other sequential structural data. In addition, the length of these sequence structure data is generally unfixed. The feedforward neural networks require that the dimensions of input and output must be fixed.

Therefore, when we are processing the sequence date, we need a new method, i.e. Recurrent Neural Networks (RNN). By using neurons with self-feedback, RNN can process sequence data in any size.

If we have an input sequence x_1: T =(x_1,x_2,...,x_t,...,x_T), RNN will renew the value h_t in embedding layer with the function as below:

$$\mathbf{h}_t = \begin{cases} 0 & t = 0 \\ f(\mathbf{h}_{t-1}, \mathbf{x}_t) & \text{otherwise} \end{cases}$$

Then we will get an approximate mathematical dynamic system that can change over time.

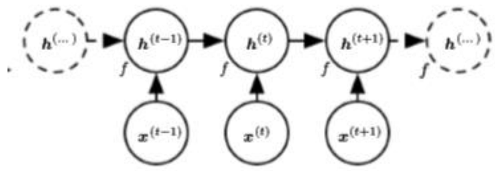
2. Models

a) SCR

Assuming at time t, we have an input x_t, embedding state h_t, we can build an RNN cell with function as below:

$$\mathbf{z}_t = U\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b},$$
$$\mathbf{h}_t = f(\mathbf{z}_t) = f(U\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b}),$$

z_t is the input with weights, bias and information of front layers (h_{t-1}). At the same time, h_t will be updated:



The information transmits in h_t and the weights are shared. We can use BPTT to generate gradient. However, when the input sequence is long, there will be a gradient explosion problem, also called long-term dependence problem. To solve the problem, we need to improve RNN.

b) LSTM-RNN

LSTM is the abbreviation of Long Short-Term Memory Neural Network, which is the most efficient improvement in RNN.

LSTM has a memory unit to save the historical information; in addition, it uses Gating Mechanism with three gates to operate the information in memory unit:

$$\mathbf{i}_t = \sigma(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + \mathbf{b}_i),$$
$$\mathbf{f}_t = \sigma(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + \mathbf{b}_f),$$
$$\mathbf{o}_t = \sigma(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + \mathbf{b}_o),$$

Use i_t (input gate) and f_t (forgotten gate) to renew memory unit with function as below:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t,$$

Then use o_t (output gate) to update embedding state h_t:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

Finally, the long-term memory can be generated and operated automatically in RNN.

c) GRU-RNN

GRU means Gated Recurrent Unit, which has two gates:

$$\mathbf{r}_t = \sigma(W_r\mathbf{x}_t + U_r\mathbf{h}_{t-1} + \mathbf{b}_r),$$
$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z),$$

r_t and z_t are reset gate and update gate: reset gate mainly controls historical information and update gate mainly controls new information.

The process of updating h_t is as below:

$$\tilde{\mathbf{h}}_t = \tanh(W_c\mathbf{x}_t + U(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}),$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t,$$

When r=0 z=0 the historical information is all forgotten.

Because GRU has only 2 gates, it improves the efficiency of operation.

3. Explanation of program

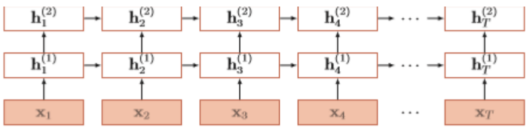
At the beginning, I process the poems, transforming the Chinese word to integer and making a dictionary. Before training, I split the poems to batches. Each batch has 64 poems.

a) Training model

I have three models in the program. But limited by computer, I only tried LSTM model. The difference among models has been interpreted in the preceding text.

Each neuron has 128 units, which also means the size of shared weights is 128.

I use “MultiRNNCell” to make a stacked recurrent neural network, which has two embedding layers like this:



After I transform the poems to integer, the dimensions of information are so large that I cannot input it to RNNcells, so I use an embedding layer to narrow down the dimensions.

“tf.nn.dynamic_rnn” is the process of updating h_t and generate outputs. I use one_hot to operate the real output date. I use a full-connection layer to translate the predicted output after which I also use softmax to gain the probability of predicted outputs. Then, I can calculate the loss with cross entropy, and use Adam to diminish the gradient.

b) Training process

Train with learning rate equals 0.01.

Because I need to use the results of first training process on the bigger dataset when I ran training on the smaller one, I merged the two dictionaries.

c) Generate poems

The prediction is the labels of words. With the generated dictionary, I can map the prediction to a word and then generate a poem.

4. Results

a) Dataset "poems.txt"

```
################################
日日风光动，清风吹落花。
不知何处去，不井一年时。
不是春风起，岭堪此地同。
不知春草色，不觉故园春。
不觉春风起，还将白鹭赴。
不知何处去，不见此时情。
################################
restore finished
################################
红尘不可见，不见白云鹤。
不是人心在，岭堪不得知。
不知何处去，不觉一年来。
不见青山下，还将白鹭赴。
不知何处去，不见白云鹤。
不是无人问，何如此地同。
################################
restore finished
################################
山中山水上，山水有新诗。
不得寻僧话，闲吟不得闲。
闲吟一杯酒，一绽酒杯中。
白发无人到，闲吟不可寻。
不知人不见，不觉老人闲。
莫道无人说，何鳞不得知。
################################
```

```
################################
夜夜夜夜雨，一声秋声扇。
月明夜夜月，月明夜夜岫。
月明秋月冷，月明月照明。
月明秋月冷，月明秋水流。
夜深愁不去，畔复不如梦。
愁思愁怨断，愁怨泪胡泪。
################################
restore finished
################################
湖上一年别，一杯芳草生。
不知何处去，不觉此时新。
不是春风起，岭堪此地同。
不知何处去，不觉此时来。
不觉风尘好，还应不可寻。
不知何处去，不见白云鹤。
################################
restore finished
################################
海上山河上，东南望远游。
山河通海气，山水入云中。
山水幽天阙，山河入洞门。
山河通海气，山水入云深。
日暮云中远，风清月色明。
此中无檐事，何壁此时同。
################################
```

```
################################
restore finished
################################
月明清夜月，月明清漏声。
清光摇落和，荃漾馆清光。
不井风光动，还同白露繁。
不知风和止，不觉此时情。
不觉风光动，还随白露飘。
不知何处去，不井此时同。
################################
```

b) Dataset "tangshi.txt"

```
################################
日墙力河深，然能一江诏。
在唯动近酒，从迷缕拜千。
我戟泛更约，荒阴经舆严。
生尽细偷与，春知欲成作。
################################
restore finished
################################
红照丝悲持占，寂变路千。
################################
restore finished
################################
龙此生关旧，苍苍事青侯玉。
颜白不上年明，心三疑不将疑。
西人山因，州何水深开还。
映东不人窗将因直，断青州处觉。
游水向向，谢况荷露，飞回尽合生春。
蝉蝉两叶地，生尽细偷与。
知上鬓，曾好青畔戎，残花好落街。
问运问时日违，至镜永日意。
绿浮二不君拟黻，场时寄有世远。
################################
```

```
################################
restore finished
################################
逐树少少少头，只只隔因大大。
################################
restore finished
################################
间残下尺如明，作心楚身不期。
三北宜莺三城和惊，笼已已君不向。
################################
restore finished
################################
勋接接穿穿起，雅苦程沈道。
################################
restore finished
################################
月裴多芦起身怅，怅闭枝打满晚寒。
知稀凉手索偏丞，中玉事白翔夜秋。
################################
```

c) Summary

According to the results, we can see that generated poems from the first dataset is neater than the ones from the second dataset, because the numbers of words of each sentence from the poems are similar. LSTMRNN cell is good at learning the termination of sentences, so that if the poem dataset is neat, the result will be neat as well.

The generated poem has some repeated words and similar sentences, which can still be identified as Machine's. Maybe because the dataset is not big enough.

Although I only tried LSTM model, I expect that the RNN model will diverse the words in and between sentences, and the length of sentences. The results of gru model will not be better than the LSTM model.