Report on Project 3

Zhunxuan Wang, 13300180086
School of Mathematical Sciences

November 28, 2017

1 Recurrent Neural Network

1.1 Basic RNN Cell

RNN is an efficient model to deal with sequential information. For each node in the same layer, *recurrent* means they perform the same task for every element in the sequence, with the output being depended on the previous computations (memory) [3]

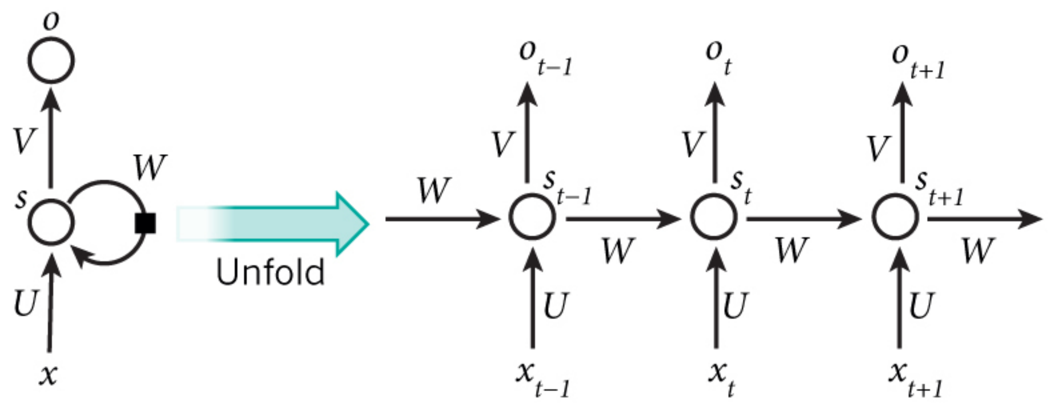


Figure 1: The RNN and its Unfolding

The above diagram shows the unfolding of basic RNN, of which each node shares the same parameters. The formulas dominating the calculation are expressed as follows

- $\mathbf{x}_t$ is the input vector (i.e. word embedding) at time step t.
- $\mathbf{s}_t$ is the hidden state (memory) at time step t, which is dominated by the previous hidden state and the input vector
$$\mathbf{s}_t = f\left(\mathbf{W}\mathbf{s}_{t-1} + \mathbf{U}\mathbf{x}_t\right)$$
where f is generally a non-linear function (e.g. tanh, `ReLu`)
- $\mathbf{o}_t$ is the output vector, which is dominated by the current hidden state
$$\mathbf{o}_t = g\left(\mathbf{V}\mathbf{s}_t\right)$$
where g depends on what we need (e.g. `softmax` for obtaining the probability of each words).
- $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ are the shared parameters, which are the parameters of a basic RNN cell.

1.2 Basic LSTM Cell

Long short-term memory (LSTM) [2] is a special kind of RNN, which is capable of remember long term dependencies. A basic LSTM cell is shown as follows

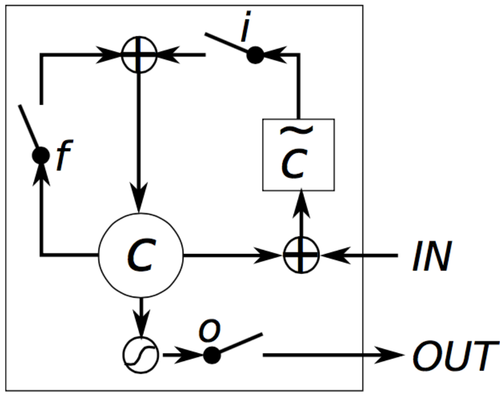


Figure 2: Basic Structure of an LSTM Cell

As the diagram shows, we introduce $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, and $\mathbf{c}_t$, $\tilde{\mathbf{c}}_t$ of which the definition and the formula are expressed as follows

- $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the input, forget and output gates respectively. They have the same form but different parameters

$$\begin{aligned}\mathbf{i}_t &= \sigma\left(\mathbf{W}_{\mathrm{i}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{i}}\mathbf{x}_t\right)\\ \mathbf{f}_t &= \sigma\left(\mathbf{W}_{\mathrm{f}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{f}}\mathbf{x}_t\right)\\ \mathbf{o}_t &= \sigma\left(\mathbf{W}_{\mathrm{o}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{o}}\mathbf{x}_t\right)\end{aligned}$$

  where $\mathbf{s}_t$, $\mathbf{x}_t$ are the hidden state and the input vector at time step t, $\mathbf{W}_{\mathrm{i,f,o}}$ and $\mathbf{U}_{\mathrm{i,f,o}}$ are the parameters of each gate.

- $\tilde{\mathbf{c}}_t$ is the new candidate value, which could be added to the state

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_{\mathrm{c}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{c}}\mathbf{x}_t\right)$$

  of which the form is similar to the gates, but with a different activation function tanh.

- $\mathbf{c}_t$ is the internal memory, which is the sum of previous memory $\mathbf{c}_{t-1}$ multiplied by the forget gate $\mathbf{f}_t$, and the candidate value $\tilde{\mathbf{c}}_t$ multiplied by the input gate $\mathbf{i}_t$

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

  where $\circ$ is the operator of elementwise muliplication.

- $\mathbf{s}_t$ is obtained by

$$\mathbf{s}_t = \tanh\left(\mathbf{c}_t\right) \circ \mathbf{o}_t.$$

Thereby the structure of a basic LSTM cell is established.

1.3 Basic GRU Cell

The structure behind a basic GRU (Gated Recurrent Unit) cell is quite similar to that of a basic LSTM cell

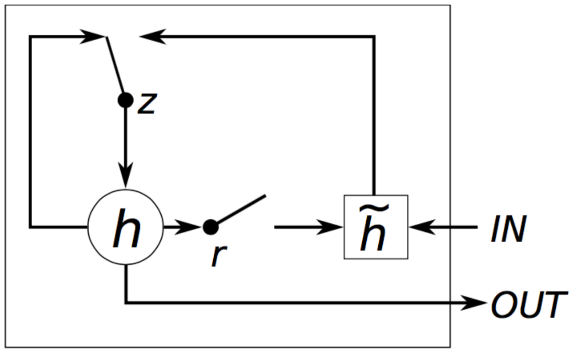


Figure 3: Basic Structure of a GRU Cell

As the diagram shows, we introduce $\mathbf{z}_t$, $\mathbf{r}_t$, and $\mathbf{h}_t$, $\tilde{\mathbf{h}}_t$ of which the definition and the formula are expressed as follows [1]

- $\mathbf{z}_t$, $\mathbf{r}_t$ are the update and reset gates respectively. They have the same form but different parameters

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{\mathrm{z}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{z}}\mathbf{x}_t\right)$$
$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{\mathrm{r}}\mathbf{s}_{t-1} + \mathbf{U}_{\mathrm{r}}\mathbf{x}_t\right)$$

  where $\mathbf{s}_t$, $\mathbf{x}_t$ are the hidden state and the input vector at time step t, $\mathbf{W}_{\mathrm{z,r}}$ and $\mathbf{U}_{\mathrm{z,r}}$ are the parameters of each gate.

- $\tilde{\mathbf{h}}_t$ is the new candidate activation

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_{\mathrm{h}}\left(\mathbf{s}_{t-1} \circ \mathbf{r}_t\right) + \mathbf{U}_{\mathrm{h}}\mathbf{x}_t\right).$$

  where $\circ$ is the operator of elementwise muliplication.

- $\mathbf{s}_t$ is obtained by

$$\mathbf{s}_t = \mathbf{s}_{t-1} \circ \mathbf{z}_t + \tilde{\mathbf{h}}_t \circ \left(\mathbf{1} - \mathbf{z}_t\right).$$

Thereby the structure of a basic GRU cell is established.

1.4 `Seq2seq` Generator

To generate poems using RNN model, we chose to adopt `seq2seq` model [4], of which the main idea of this model is to feed the current input by the previous output. We adopted 2-layer RNN model to generate poems

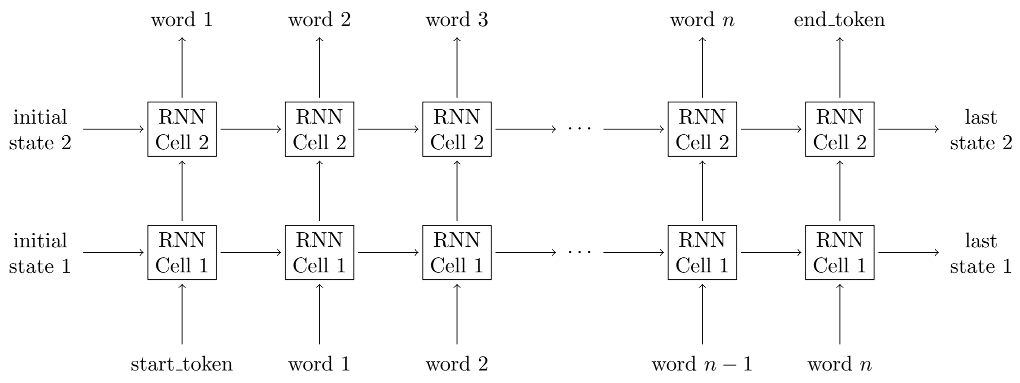


Figure 4: An Example of Poems Generating

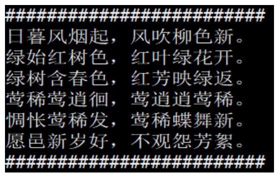
As we observe from the graph above, the generating began with feeding input by a start token. After that, the input of each iteration would be fed by the previous output, and the incoming state would be the previous state. When it came to the last word, the generating would end with an end token.

It is worth mentioning that, if we specify a word to begin, the input after the start token will be the specific word.

2 Experimental Results

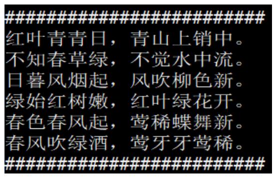
Setting the loss function the cross entropy of prediction and one-hot target, choosing LSTM as the RNN cell, the 2-layer RNN model was trained after 50 epochs, based on the dataset `poems.txt`.

we specified 8 words to start with, and the poems generated are shown as follows



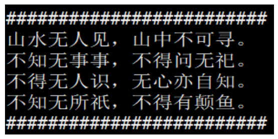
```
##########################
日暮风烟起，风吹柳色新。
绿始红树色，红叶绿花开。
绿树含春色，红芳映绿返。
莺稀莺逍徊，莺逍逍莺稀。
惆怅莺稀发，莺稀蝶舞新。
愿邑新岁好，不观怨芳絮。
##########################
```

(a) 日



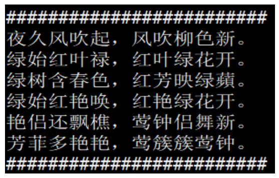
```
##########################
红叶青青日，青山上销中。
不知春草绿，不觉水中流。
日暮风烟起，风吹柳色新。
绿始红树嫩，红叶绿花开。
春色春风起，莺稀蝶舞新。
春风吹绿酒，莺牙牙莺稀。
##########################
```

(b) 红



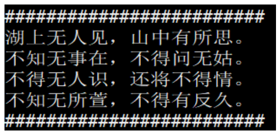
```
##########################
山水无人见，山中不可寻。
不知无事事，不得问无祀。
不得无人识，无心亦自知。
不知无所祇，不得有颠鱼。
##########################
```

(c) 山



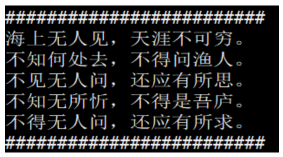
```
##########################
夜久风吹起，风吹柳色新。
绿始红叶禄，红叶绿花开。
绿树含春色，红芳映绿蘋。
绿始红艳唤，红艳绿花开。
艳侣还飘樵，莺钟侣舞新。
芳菲多艳艳，莺簇簇莺钟。
##########################
```

(d) 夜



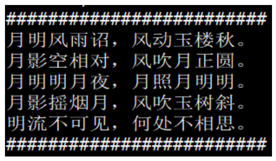
```
##########################
湖上无人见，山中有所思。
不知无事在，不得问无姑。
不得无人识，还将不得情。
不知无所萱，不得有反久。
##########################
```

(e) 湖



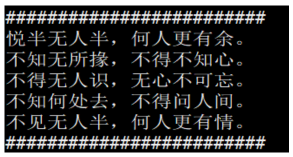
```
##########################
海上无人见，天涯不可穷。
不知何处去，不得问渔人。
不见无人问，还应有所思。
不知无所忻，不得是吾庐。
不得无人问，还应有所求。
##########################
```

(f) 海



```
##########################
月明风雨诏，风动玉楼秋。
月影空相对，风吹月正圆。
月明明月夜，月照月明明。
月影摇烟月，风吹玉树斜。
明流不可见，何处不相思。
##########################
```

(g) 月



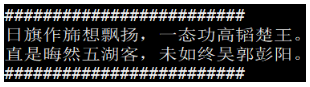
```
##########################
悦半无人半，何人更有余。
不知无所掾，不得不知心。
不得无人识，无心不可忘。
不知何处去，不得问人间。
不见无人半，何人更有情。
##########################
```

(h) 悦

Figure 5: LSTM-Generated Poems Based on `poems.txt`

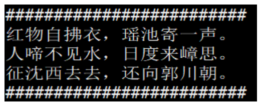
As we observe from the poems above, they are all five-character poems, and the statement of the generated poems is relatively fluent. However the logic in the context did not make much sense, and too many words repeated.

Retraining the model based on the previous model and the new `tangshi.txt` dataset, a new set of poems were generated, as the figures above show



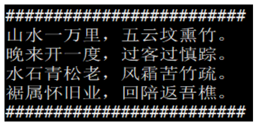
```
#########################
日旗作旆想飘扬，一态功高韬楚王。
直是晦然五湖客，未如终吴郭彭阳。
#########################
```

(a) 日



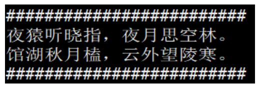
```
#########################
红物自拂衣，瑶池寄一声。
人啼不见水，日度来嶂思。
征沈西去去，还向郭川朝。
#########################
```

(b) 红



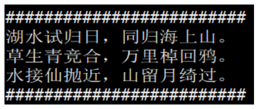
```
#########################
山水一万里，五云坟熏竹。
晚来开一度，过客过慎踪。
水石青松老，风霜苦竹疏。
裾属怀旧业，回陪返吾樵。
#########################
```

(c) 山



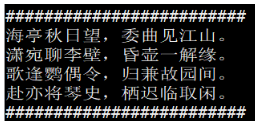
```
#########################
夜猿听晓指，夜月思空林。
馆湖秋月榼，云外望陵寒。
#########################
```

(d) 夜



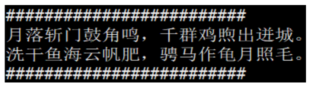
```
#########################
湖水试归日，同归海上山。
草生青竞合，万里棹回鸦。
水接仙抛近，山留月绮过。
#########################
```

(e) 湖



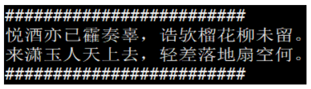
```
#########################
海亭秋日望，委曲见江山。
潇宛聊李壁，昏壶一解缘。
歌逢鹦偶令，归兼故园间。
赴亦将琴史，栖迟临取闲。
#########################
```

(f) 海



```
#########################
月落斩门鼓角鸣，千群鸡煦出进城。
洗干鱼海云帆肥，骋马作龟月照毛。
#########################
```

(g) 月



```
#########################
悦酒亦已霍奏辜，诰欤榴花柳未留。
来潇玉人天上去，轻差落地扇空何。
#########################
```

(h) 悦

Figure 6: LSTM-Generated Poems Based on `poems.txt` and `tangshi.txt`

As we observe from the poems above, seven-character poems appeared, and the statement of the generated poems and the logic in the context is more fluent than before. Furthermore, the number of repeating words reduced a lot, and the efficiency on antithesis improved. In general, the model based on `poems.txt` and `tangshi.txt` is more efficient than that based on only `poems.txt`.

3 Comparison between basicRNN, LSTM and GRU

LSTM and GRU are gated mutants of RNN structure. They all perform well on poem generator model training, and better than basic RNN structure, with reference to the total train loss mean in the last epoch (when reaching the convergence state) L (as the following table shows, only on `poems.txt`)

	basicRNN	LSTM	GRU
L	5.819501	4.082350	4.222888

From the table we can infer that the performance by LSTM and GRU are close (LSTM are a little better than GRU), each of which performs better than basic RNN (on `poems.txt`).

The poems generated by basic RNN and GRU are in the appendix A. As we can obtain from the poems, the poems generated by basic RNN are very short and monotonous, and the poems generated by GRU make more sense that by basic RNN, but they lack more logic in context and have more repeating words than LSTM-generated poems.

References

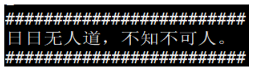
[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

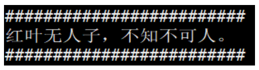
[4] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

A Poems by basicRNN and GRU



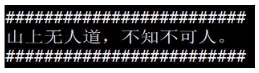
```
##########################
日日无人道，不知不可人。
##########################
```

(a) 日



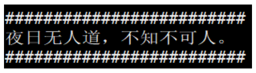
```
##########################
红叶无人子，不知不可人。
##########################
```

(b) 红



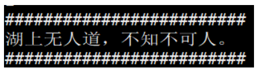
```
##########################
山上无人道，不知不可人。
##########################
```

(c) 山



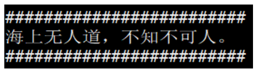
```
##########################
夜日无人道，不知不可人。
##########################
```

(d) 夜



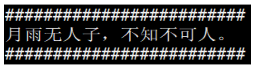
```
##########################
湖上无人道，不知不可人。
##########################
```

(e) 湖



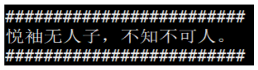
```
##########################
海上无人道，不知不可人。
##########################
```

(f) 海



```
##########################
月雨无人子，不知不可人。
##########################
```

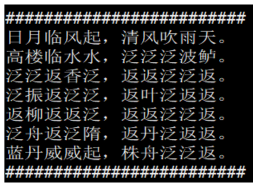
(g) 月



```
##########################
悦袖无人子，不知不可人。
##########################
```

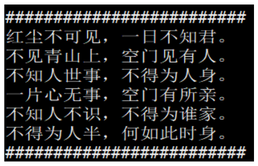
(h) 悦

Figure 7: basicRNN-Generated Poems Based on `poems.txt`



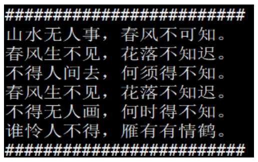
```
##########################
日月临风起，清风吹雨天。
高楼临水水，泛泛泛波鲈。
泛泛返香泛，返返泛泛返。
泛振返泛泛，返叶泛返返。
返柳返返泛，返返泛泛返。
泛舟返泛隋，返丹泛返返。
蓝丹威威起，株舟泛泛返。
##########################
```

(a) 日



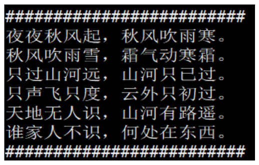
```
##########################
红尘不可见，一日不知君。
不见青山上，空门见有人。
不知人世事，不得为人身。
一片心无事，空门有所亲。
不知人不识，不得为谁家。
不得为人半，何如此时身。
##########################
```

(b) 红



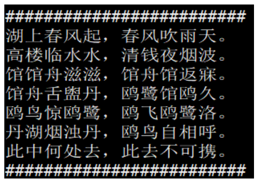
```
##########################
山水无人事，春风不可知。
春风生不见，花落不知迟。
不得人间去，何须得不知。
春风生不见，花落不知迟。
不得无人画，何时得不知。
谁怜人不得，雁有有情鹤。
##########################
```

(c) 山



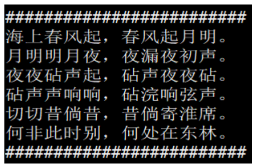
```
##########################
夜夜秋风起，秋风吹雨寒。
秋风吹雨雪，霜气动寒霜。
只过山河远，山河只已过。
只声飞只度，云外只初过。
天地无人识，山河有路遥。
谁家人不识，何处在东西。
##########################
```

(d) 夜



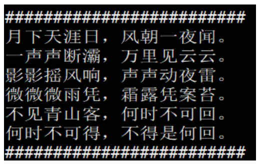
```
##########################
湖上春风起，春风吹雨天。
高楼临水水，清钱夜烟波。
馆馆舟滋滋，馆舟馆返寐。
馆舟舌盥丹，鸥鹭馆鸥久。
鸥鸟惊鸥鹭，鸥飞鸥鹭洛。
丹湖烟浊丹，鸥鸟自相呼。
此中何处去，此去不可携。
##########################
```

(e) 湖



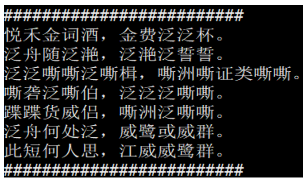
```
##########################
海上春风起，春风起月明。
月明明月夜，夜漏夜初声。
夜夜砧声起，砧声夜夜砧。
砧声声响响，砧浣响弦声。
切切昔倘昔，昔倘寄淮席。
何非此时别，何处在东林。
##########################
```

(f) 海



```
##########################
月下天涯日，风朝一夜闻。
一声声断灞，万里见云云。
影影摇风响，声声动夜雷。
微微微雨凭，霜露凭案苔。
不见青山客，何时不可回。
何时不可得，不得是何回。
##########################
```

(g) 月



```
##########################
悦禾金词酒，金费泛泛杯。
泛舟随泛滟，泛滟泛誓誓。
泛泛嘶嘶泛嘶楫，嘶洲嘶证类嘶嘶。
嘶砻泛嘶伯，泛泛泛嘶嘶。
蹀蹀货威侣，嘶洲泛嘶嘶。
泛舟何处泛，威鹭或威群。
此短何人思，江威威鹭群。
##########################
```

(h) 悦

Figure 8: GRU-Generated Poems Based on `poems.txt`