CEDL HW1

104062572 林暐翔，104062599 李昶輝

Selcected Paper : An Empirical Exploration of Recurrent Network Architectures
Author : Rafal Jozefowicz, Wojciech Zaremba, Ilya Sutskever

Introduction

A criticism of the LSTM architecture is that it is ad-hoc and that it has a substantial number of components whose purpose is not immediately apparent. As a result, it is also not clear that the LSTM is an optimal architecture, and it is possible that better architectures exist.

This paper has done an architecture search over ten thousand different RNN architectures, in order to verifying if there exist any architecture better than LSTM. After trying the thousand architectures, they identified an architecture similar to Gated Recurrent Unit(GRU) outperforms both LSTM and GRU on most tasks.

They also performed experiments to measure the importance of each of the LSTM's components. They discovered that input gate is important and output gate is unimportant. Forget gate is significant on all problems except language modelling.

Also, they emphasize that adding a bias of 1 to the LSTM's forget gate will make its performance close to GRU.

Search Method

They defined a procedure to stochastic mutate an architecture based on LSTM or GRU and randomly initialize such architecture to evaluate its performance. If the architecture pass the simple memorization problem, then following steps might be applied to it.

They optimize architectures which may have potential with three different task, "Arithmetic", "XML modeling" and "Penn Tree-Bank". Finally, evaluate these architectures with task "Music".

After trying ten thousand architectures, they maintained a list of 100 best architectures.

Best Architecture discovered

MUT1:

$$
\begin{aligned}
z &= \mathrm{sigm}(W_{\mathrm{xz}} x_t + b_{\mathrm{z}}) \\
r &= \mathrm{sigm}(W_{\mathrm{xr}} x_t + W_{\mathrm{hr}} h_t + b_{\mathrm{r}}) \\
h_{t+1} &= \tanh(W_{\mathrm{hh}}(r \odot h_t) + \tanh(x_t) + b_{\mathrm{h}}) \odot z \\
&+ h_t \odot (1 - z)
\end{aligned}
$$

MUT3:

$$
\begin{aligned}
z &= \mathrm{sigm}(W_{\mathrm{xz}} x_t + W_{\mathrm{hz}} \tanh(h_t) + b_{\mathrm{z}}) \\
r &= \mathrm{sigm}(W_{\mathrm{xr}} x_t + W_{\mathrm{hr}} h_t + b_{\mathrm{r}}) \\
h_{t+1} &= \tanh(W_{\mathrm{hh}}(r \odot h_t) + W_{xh} x_t + b_{\mathrm{h}}) \odot z \\
&+ h_t \odot (1 - z)
\end{aligned}
$$

Arch.	Arith.	XML	PTB
Tanh	0.29493	0.32050	0.08782
LSTM	0.89228	0.42470	0.08912
LSTM-f	0.29292	0.23356	0.08808
LSTM-i	0.75109	0.41371	0.08662
LSTM-o	0.86747	0.42117	0.08933
LSTM-b	0.90163	0.44434	0.08952
GRU	0.89565	0.45963	0.09069
MUT1	**0.92135**	**0.47483**	0.08968
MUT2	0.89735	**0.47324**	0.09036
MUT3	0.90728	0.46478	**0.09161**

Table 1. Best next-step-prediction accuracies

They discoverd 3 different architecture mutations. MUT1 & MTU3 are good at different task. But it is not easily to see why MUT3 is good at PTB task from its mathematical formula directly. And it is very likely to have some better architeture.

Discussion

The architecture they found, “MUT1”, matched the GRU’s performance on language modeling and better than GRU on all other tasks. So it might be a good choice on those task they have tested.

If you prefer to use LSTM, two techniques might help to improve the performance.

1) using Dropout
2) apply larger bias on forget gate

And if you want to reduce parameters on LSTM, according to their experiment, output gate might be a good choice to remove.

We could try diffrent models to solve our problem for best performance. But basic LSTM with bias & GRU seems to work fine. So maybe it is more important to utilize RNNs, i.e. using new stack, network topology.

In newst Google’s Neural Machine Translation System (GNMT), it use LSTM with risidual connection. It may worth searching to see whether there is a better RNN model for the task because LSTM may not be the perfect one.