Simple Gated Unit and Its Deep Version

Yuan Gao

University of Helsinki

gaoyuankidult@gmail.com

September 22, 2015

Overview

1. RNN Conventional Graph
   - RNN Conventional Graph Rules
   - RNN Conventional Graph Examples

2. Simple Gated Unit(SGU) and Deep SGU
   - Recurrent Gated Unit
   - Simple Gated Unit(SGU)
   - Deep SGU (DSGU)
   - Experiments

RNN Conventional Graph

We describe different recurrent neuron network using a standard form of graph, by which, we call RNN Convention Graph(RCG). A RCG can show detailed inner structure of RNN i.e. relationship between input and output of each time step.

Basic Idea

Normally, RCG takes x_t (sometimes also x_1 to x_{t_1}) and h_{t-1} (sometimes also h_1 to h_{t-2}) as inputs and results h_t as output. A RCG represents input at left side and output at right side, which enables the graph to show clearly how does information at left side flow through different structures to right side.

RNN Conventional Graph Rules

Rule 1

Arrow $\rightarrow$ means multiplication with Matrix. Activation functions are put on $\rightarrow$ or a simple line $-$ to show an activation function is applied on a transformation of a variable $\sigma(W\vec{x})$ or just on a variable $\sigma(\vec{x})$.

Rule 2

Variables are connected by operation nodes. An addition node $\bigoplus$ takes inputs from left and up side and then makes an addition operation then it output to down side(if there are only two inputs) or right side (if there are three inputs). It is similar for multiplication nodes $\bigotimes$.

Rule 3

If a special operation is needed, the definition should be included in the description.

Examples: Vanilla Recurrent Neural Network

Mathematically, updating rule of SRNN is defined as:

$$\vec{h}_t = \sigma\left(W_{xh}\vec{x} + W_{hh}\vec{h}_{t-1} + \vec{b}\right) \tag{1}$$

Graphically, it can be represented as:

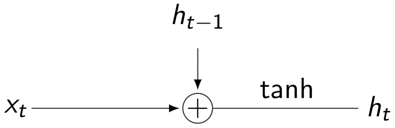


Figure : Conventional Graph of Vanilla Recurrent Neural Network

Graphically, it can be represented as:

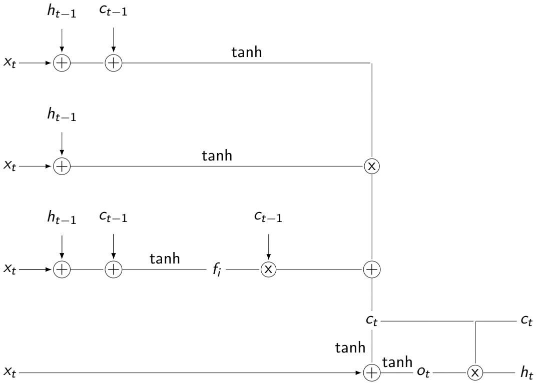


Figure : Conventional Graph of Long Short Term Memory

Examples: Long Short-Term Network

Mathematically, updating rule of SRNN is defined as:

$$\vec{i}_t = \sigma\left(W_{xi}\vec{x}_t + W_{hi}\vec{h}_{t-1} + W_{ci}\vec{c}_{t-1} + \vec{b}_i\right) \tag{2}$$

$$\vec{f}_t = \sigma\left(W_{xf}\vec{x}_t + W_{hf}\vec{h}_{t-1} + W_{cf}\vec{c}_{t-1} + \vec{b}_f\right) \tag{3}$$

$$\vec{c}_t = \vec{f}_t\vec{c}_{t-1} + \vec{i}_t \tanh\left(W_{xc}\vec{x}_t + W_{hc}\vec{h}_{t-1} + \vec{b}_c\right) \tag{4}$$

$$\vec{o}_t = \sigma\left(W_{xo}\vec{x}_t + W_{ho}\vec{h}_{t-1} + W_{co}\vec{c}_t + \vec{b}_o\right) \tag{5}$$

$$\vec{h}_t = \vec{o}_t \tanh(\vec{c}_t) \tag{6}$$

Examples: Recurrent Gated Unit

GRU is first designed by Kyunghyun Cho in his paper about Neural Machine Translation[3]. This structure of RNN only contains two gates. Update gate controls the information that flows in the memory whilst reset gate controls the information that flows out of memory.

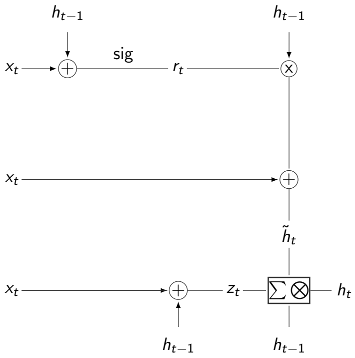


Figure : Conventional Graph of Recurrent Gated Unit

Examples: Recurrent Gated Unit

Mathematically, updating rule of GRU is defined as:
Gate $\boxed{\sum \bigotimes}$ is defined as:

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \tag{7}$$

Simple Gated Unit

A Simple Gated Unit(SGU) is a recurrent structure designed for learning long term dependences. Its aim is to reduce the amount of parameters needed for training and to accelerate training speed.

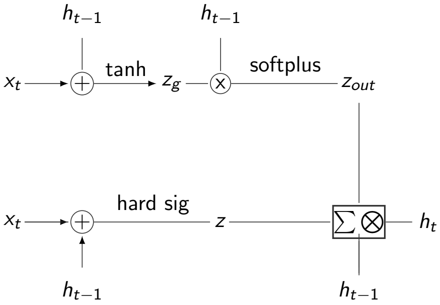


Figure : Conventional Graph of Recurrent Gated Unit

Simple Gated Unit

Result Plot:

$$x_g = W_{xh}\vec{x}_t \tag{8}$$
$$z_g = \sigma_1(x_g \cdot h_{t-1}) \tag{9}$$
$$z_{out} = \sigma_2(z_g \cdot h_{t-1}) \tag{10}$$
$$z = \sigma_3(W_{xz}x_t + W_{hz}h_{t-1}) \tag{11}$$
$$h_t = (1 - z_t)h_{t-1} + z_t \cdot z_{out} \tag{12}$$

DSGU

avoid shallowness of the gate and the nature of the softmax activation function.

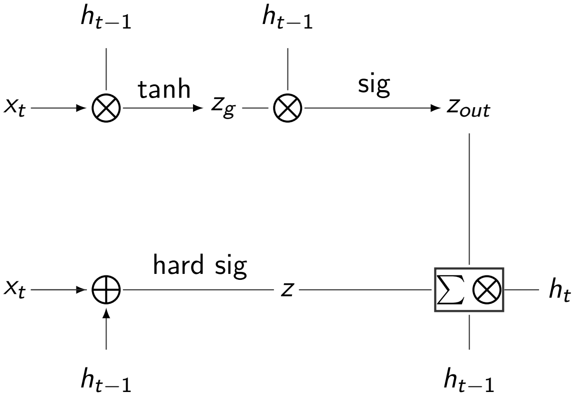


Figure : The structure of DSGU. Like in the case of SGU, the input is fed into two different function units of the structure, namely z and z_{out}.

Mathematical representation of SGU.

$$x_g = W_{xh}\vec{x}_t \tag{13}$$

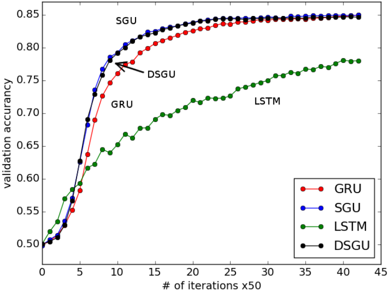
$$z_g = \sigma_1(x_g \cdot h_{t-1}) \tag{14}$$

$$z_{out} = \sigma_2(W_{go}(z_g \cdot h_{t-1})) \tag{15}$$

$$z = \sigma_3(W_{xz}x_t + W_{hz}h_{t-1}) \tag{16}$$

$$h_t = (1 - z_t)h_{t-1} + z_t \cdot z_{out} \tag{17}$$

IMDB Sentiment Analysis



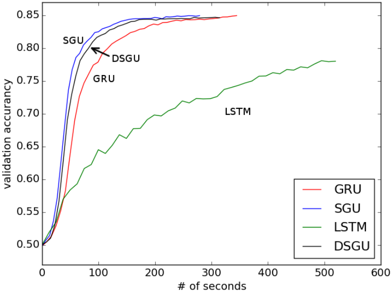


Figure : Comparison between SGU, DSGU, GRU and LSTM on the IMDB sentiment classification task in terms of time and number of iterations required to perform the task.

Pixel-by-pixel MNIST Classification

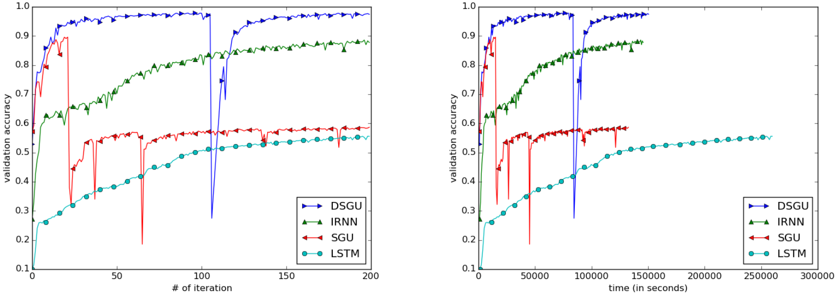


Figure : Validation accuracy of different models including DSGU, IRNN and SGU and LSTM in terms of number of iterations and time in the MNIST classification task. Both DSGU and SGU reached a very high accuracy within a short period of time. However, SGU dropped after short period, which might be due to the fact that it is too simple for learning this task.

Text Generation Task

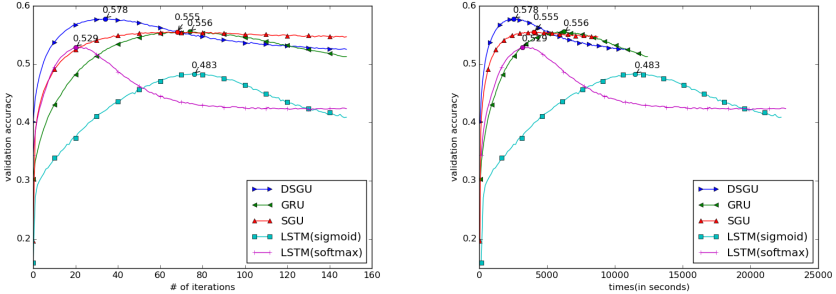


Figure : Validation accuracy of different models including DSGU, GRU and SGU and LSTM in terms of the number of iterations and time in the text generation task. Each line is drawn by taking the mean value of 15 runs of each configuration.