(DATE)　２０２４年　１２月　２日

Department of Electrical and Electronic Information Engineering	ID	M213262
Name	Takuya Betchaku	

Supervisor	Shuichi Ichikawa

Abstract

Title	Investigation of the Gate Recurrent Unit implemented with Stochastic Computing

The GRU (Gated Recurrent Unit) is a type of RNN (Recurrent Neural Network) that has been improved to better learn long-term dependencies. Its main applications include processing time-series data, such as stock price prediction and weather forecasting, as well as natural language processing tasks such as translation and sentiment analysis. Since GRU involves complex computations, its implementation consumes significant amount of hardware resources and power consumption.For IoT and mobile/edge computing, the constraints on hardware resources and power consumption have become major concerns.

In recent years, Stochastic Computing (SC) has been applied to neural network computations, demonstrating improvements in power efficiency. By simplifying arithmetic operations such as addition and multiplication, it is expected to reduce hardware costs and power consumption. Maor et al. (2019) achieved power reduction by implementing SC in LSTMs, a type of RNN, using RTL descriptions. However, latency evaluation has not been conducted, leaving its practicality unclear.

This study aims to implement SC for the GRU using C++ and high-level synthesis (HLS). A Galois-type 32-bit LFSR is employed for SC, based on the unipolar format. The power consumption and hardware cost are compared with those of LSTM.

For the evaluation, handwritten digit images from MNIST test dataset were used. The weights and biases were saved using Python library Keras and imported into C++ for inference. The accuracy of Keras model was compared to that of my C++ implementation, confirming that both achieved equivalent accuracy. Additionally, the impact of the number of GRU units on accuracy was investigated. Vitis HLS and Vivado were used for resource measurements, evaluating the resource usage of LUTs and FFs, as well as latency and power consumption. For comparison, LSTM implementation was also evaluated.Both GRU and LSTM achieved a maximum accuracy of 97%. GRU showed approximately 2.8% less resource usage compared to LSTM, while LSTM demonstrated 50% lower latency than GRU. The power consumption of GRU was 7% lower than that of LSTM.

Next, two types of SC designs were applied to GRU: SC (multiplication) and SC (multiplication + addition). The accuracy, resource usage, latency, and power consumption were evaluated. The relationship between accuracy and SN bit length (ranging from 32 to 1024) was also investigated. With SC (multiplication), an SN bit length of 512 or more achieved accuracy comparable to the conventional GRU, while SC (multiplication + addition) required an SN bit length of 1024 to achieve comparable accuracy. Resource usage increased by 13% for SC (multiplication) and by 36% for SC (multiplication + addition). Latency increased as SN bit length increased. Power consumption was reduced by 24% with SC (multiplication) and by 3% with SC (multiplication + addition). The increased power consumption in SC (multiplication + addition) was attributed to the need for multipliers and dividers in the scaling operation of SC addition. As a result, power efficiency and resource efficiency were worse for both LSTM and GRU with the SC implementation used in this study. However, since not all arithmetic units were converted to SC, it cannot be conclusively stated that GRUs are inherently unsuitable for SC.

Future challenges include the following five aspects: SC implementation of activation functions, hardware optimization, latency reduction through SN bit length adjustments, scaling techniques in high-level synthesis, and the introduction of bipolar SC format.