Investigation of AI Techniques

Student: Dominik Kiersz[1] Supervisor: Shaun Trill[2]

[1]University of Kent [2]BAE Systems PLC

Summary

To investigate **Gated Recurrent Units** (GRUs) and improve the preliminary Recurrent Neural Network (RNN) designed to detect tracks with it's past experience and present data.

Introduction

Standard Recurrent Neural Networks (RNNs) are insufficient in retaining sequence information from the past due to the **vanishing gradient problem.** Two approaches have been developed to prevent this natural memory loss. In our scenario, we implement a specialist RNN to predict tracks on a sequence of pixels given knowledge of previous sequences (Earlier image strips).

Theory and Motivation

Both LSTM *(Long Short Term Memory, Schmidhuber 1997)* and GRU *(Gated Reccurent Unit, Cho 2014)* are ways of **gating** *(Manipulating)* information to prevent loss.

Both cells utilize a **memory element(s)** that store information from many previous time-steps, filtered by mathematical constructs called **gates**. Gates that act like neurons to process information, their own weights are calibrated during training.

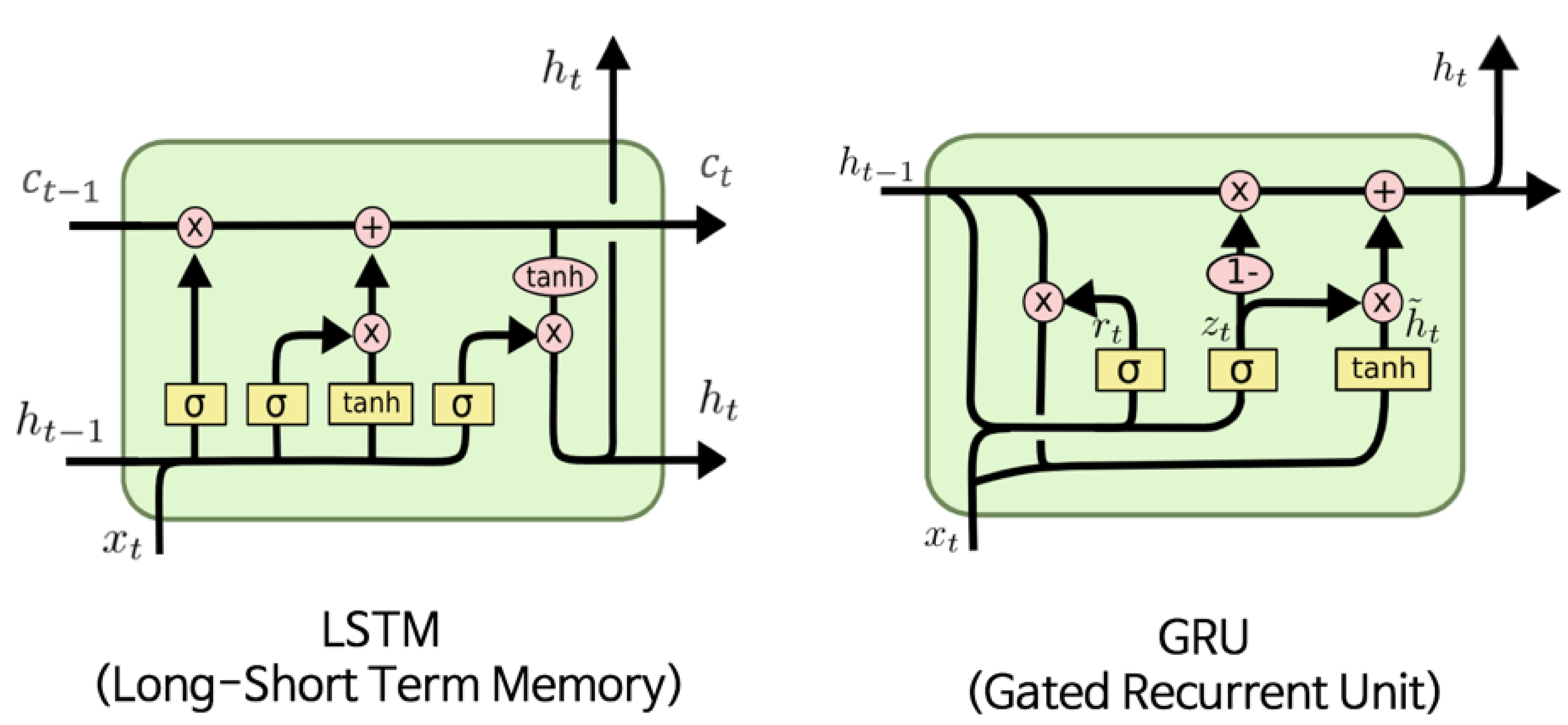

Figure: [1] The GRU and LSTM Cell

LSTMs have seen major success for the past 20 years. GRUs, however, have been motivated to provide an alternative to LSTM via a less complex mechanism - potentially saving computational power and memory.

Objectives

Given our preliminary LSTM network, the objectives were clear.

- **Test GRUs**
- **Optimize the RNN Configuration**

Methodology

With many hyper-parameters available, the following network parameters were considered and tested: Cell Type, Layer Size, Unit Number, Back-Propagation Length and network Weight Initializer.

A network with select configuration was trained in **64 hour runs**. Testing was performed by thresholding the networks sensitivity to achieve a given **% Tracks Undetected** *(TU)* and compared with other configurations for **% Tracks Detected** *(TD)* and **Detected/False Positive ratio** *(D/FP)*.

Notable Results

We have acquired many comparisons between different network setups:

- LSTMs handled better than its GRU counterpart on the preliminary configuration using the soft-sign activation function and un-normalised data.
- GRUs however vastly outperformed any currently tested confirmation with normalized data and Elu activation function (Exponential Linear Unit) on a reconfigured network.
- Above configuration improved the performance by **+10% TD and doubled the D/FP ratio** in contrast to our preliminary configuration.
- Optimized network used 2 layers of 2048 cells, utilizing the Adam optimizer at a learning rate of $1.5 * 10^{-6}$. We **balanced** the complexity of the network and the number of images passed through for best results.
- Normalising image data assisted in all know cases, converging errors within the first 10000 images to accessible levels *(Loss of* 10^{-2} *and below)*.
- Both cells generalised well, theoretically under-fitting and enabling more complex patterns to be tested in the future.

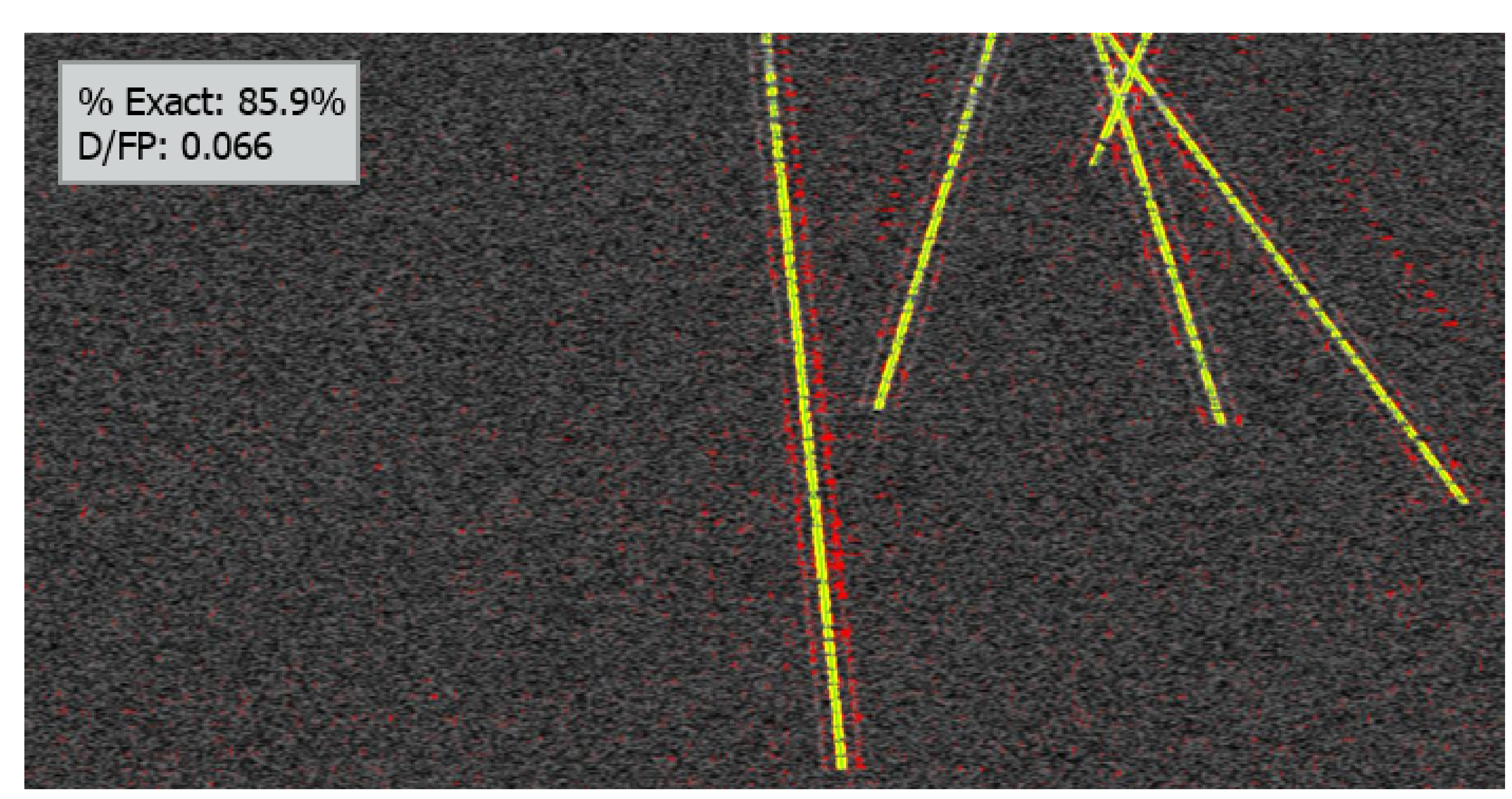

Figure: Currently the best configuration exploited, using GRU cells. *5% TU*

Future Work

Future work will involve implementation of the upgraded network to the current code enabling integration with classified software.

Within the network itself, the following may be of interest:

- HE Initializer. It is an alternative to Xavier initializer correcting halving of activation variance due to ReLU activation function. 5% increase in TU was preliminary found.
- Week-class runs on Linux machines.
- Further hyper-parameter study, following updates.

References

[1] Christopher Olah. Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015. (Accessed on 09/06/2018).