Embedded-NIST Report

By Ringgold Lin and James Fuh

Abstract

The goal of this project is to develop a trained Petri Net Neural Network that can predict the drawing of 10 integers (from 0 to 9 specifically) by using a provided myRIO board. With relatively proper data collection methods, our group successfully implemented such function with a quite high recognition rate – All 10 numbers recognizable and 7 of them with over 80% of recognition rate (based on a test of 10 draws/each over all of the 10 integers). To achieve such result, several approaches were carried out. In this report, we are going to provide a detailed explanation about the logic behind each method.

A Brief Introduction about Petri Net Neural Network

The Petri Net Neural network used in this project is a learning calculation machanisim that involves inputs, hidden layer, two weights matrices, and outputs. With proper raw data as referencing, we can acquire the two corresponding weights matrices with certain iterations of training. If the referencing raw data are representive and suitable enough, the trained network should be able to provide educated prediction of certain event and display it as the one-hotcoded output. One outline of such network is shown in *Figure 0*.

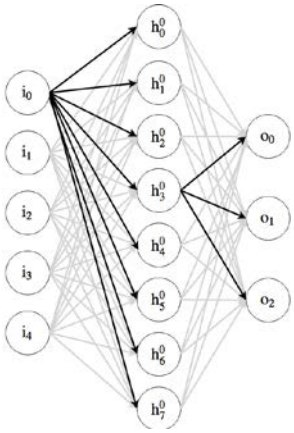

Figure 0. An example of Petri Net Neural network.

In this project, the event that is needed to be guessed is: "What is the number that was just draw?". The motive is to generate the best possible weights matrices with reasonably defined inputs and outputs.

I. *Implementation & Modification of LabView Files*

Our final implemented LabView Files was inherited then modified from the provided good sample of Lab 4.

In the original file, the inference component was designed to instantaneously determine the target (result) in a real-time based data input; similarly, the data collection function was also designed to set a favored target for each individual set of position value. An example of such data collection can be seen from the following *Figure 1*.

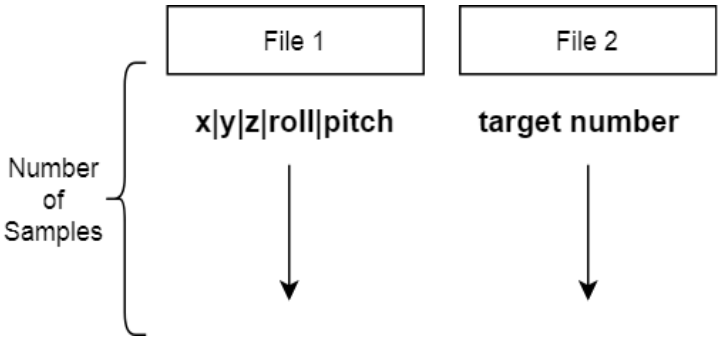

Figure 1. Original Lab 4 inherited data format. Each line of File 1 records the position value at one moment, and File 2 records the target value in that moment of range {0, 1, 2}

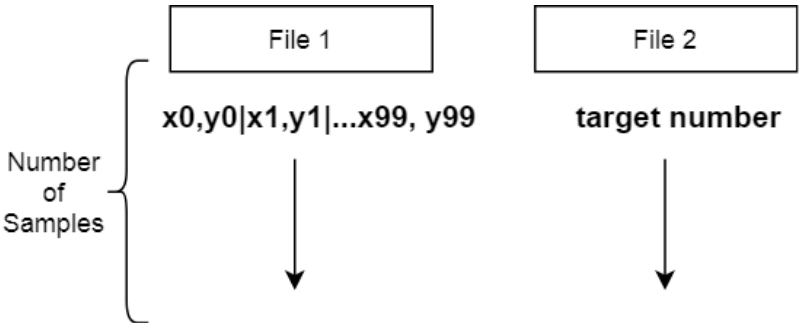

Figure 2. The new data format. File 1 records 100 sets of position values for each "draw" of integer. And File 2 records the corresponding integer (from 0 to 9)

To modify such template so that it can collect a whole "picture" set of data, we changed the amount of data collectable to be a "period" instead of a "moment". To do this, the array construction of the input data was modified as "grouped position value moments with one next to each other in one line" rather than "one set of

position value at each moment on each line". Moreover, corresponding integer result was hardcoded after each data collection of one integer draw. An example of such data format can be summarized as above in *Figure 2*.

Because of the new data format design, the overall Petri Net input and output also switched from (5, 3) to (200, 10), where the 200 stands for 100 position records of x & y values, and 10 stands for the 10 possible outcome of the integer drawing (the input parameter selection will be explained in the later section).

At this point, all of the tools of this project were set since the exact same Forward Pass and Backward Pass algorithms were used. The next question is: How to collect the reference raw data needs to be trained.

II. The First Simple Approach – Just Draw and Train

By following the experience from previous lab, we started with the most basic way of data creation: Draw the data in the air for several times then train the weights matrices until the mean square error is relatively low and not improving. The ways of each integer's drawing is shown in *Figure 3*.

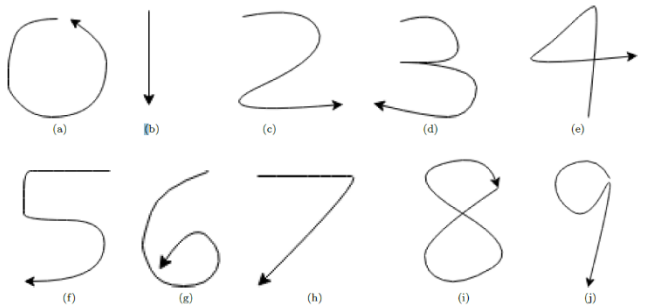

Figure 3. The normal (default) form integer drawing methods

At first, we simply drew each integer 5 times from 3 to 5 and started the training. Each integer drawing was set to be five seconds, and 500 pairs of x & y position values are collected for each integer. The goal of this trial has two purposes:

- *See if our LabView actually works, since it should be able to at least distinguish between 3 integers*
- *Get an initial feeling about the drawing and training process*

In this trial, the result of the recognition was promising: The network was able to identify all three integers correctly. However, the recognition rate of 5 and 3 are only about 50% (they were confusing with each other). This is probably because of the similarity at the second part of their drawing, where the lower part of 5 and 3 are exactly the same. Moreover, we found out that the x & y values on the accelerometer graph is quite shaky rather than stable. Such shakiness is because of the long drawing time and over generated data sets, this makes the slight shaking of hand affects the data collection for a great deal. Thus, we plan to make following improvements to the drawing:

- *Decrease the drawing time from 5 seconds to 2 seconds (prevent hand shaking effect)*
- *Decrease the pairs of x & y for each integer of drawing from 500 to 100 (prevent hand shaking effect)*
- *Increase the sample amount for each integer from 5 to 10 (increase the training generosity of each individual integer and the precision of overall identification)*
- *Make the drawing of each integer to be big enough and quick enough, so that the features of each number are different enough (increase precision of each individual identification)*

The second trial was immediately made with all of the 10 integers and the improvements summarized above, the resulting identification accuracy is listed under the table below (each number was tested for 10 times).

Trial	0	1	2	3	4	5	6	7	8	9
Success out of 10	4	3	6	4	8	6	3	9	7	5

To be more specific, 0, 6 and 9 keep confuse with each other. The few good numbers are 4, 5, 7, 8 and 9. Up to this point, we conclude from the above observation that: The numbers who have similar ways of drawing may conflict with each other (e.g. 0 and 6 have quite similar drawing curve). While the numbers who have unique (different) ways of drawing tend to be more recognizable. Thus, for the further improvement, we come with the thinking: How can we make sure every individual one of the numbers are unique? This is where we introduce the Block Form Drawing.

III. Block Form Drawing

MyRIO board collects the acceleration data instead of the location value within a time period. This is the reason we set the drawing period of any individual number to be within 2 seconds, because in this way, the acceleration can be much better highlighted (a quicker draw results in a bigger acceleration). However, the uniqueness of each number does not necessarily improve with better acceleration according to the

previous data collection and test result. Thus, we redesigned the whole methods of how to draw each individual numbers so that every single one of them is different and the similarities are minimized.

Block Form Drawing (as shown in *Figure 4*) is a newly designed way of drawing. The orange line

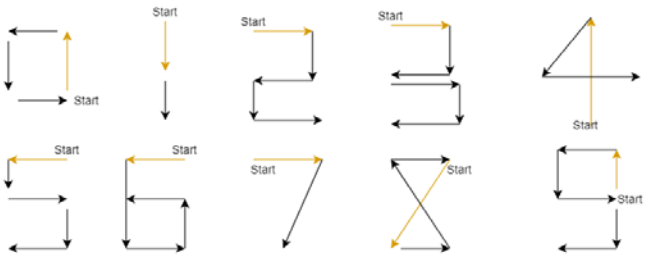

Figure 4. A visual example of the Block Form Drawing

represents the starting draw. The tail states the point that drawing start to accelerate and the head states the point that drawing starts to decelerate. The logics and theories behind such form of drawing can be summarized as follow:

- *Since every number has been defined with quite unique feature, the acceleration and deceleration should differentiate each number in a greater degree*
- *Because of the frequent acceleration and deceleration, the acceleration collected by the myRIO board should be much clearer*
- *Because of the regulated shape of each number, theoretically, the collected data of each number should be more similar, thus reduce the potential shakiness of human's hand*

With all above redesigned and setup, we recollected the data by drawing in the air. The mean square error was again waited until 0.137 where the value itself stopped to reduce for about 30 seconds. The tested drawing table for the generated weights matrices are shown below:

Trial	0	1	2	3	4	5	6	7	8	9
Success out of 10	2	0	1	2	5	3	2	4	0	2

The result is relatively disappointing since none of the testing result for any single number is better than the previous result. After a detailed analysis on the data we collected, we find out that the timing tolerance of the block form drawing is way lower than the normal way of drawing. Such example can be observed from *Figure 5.1* and *Figure 5.2*.

The orange color line indicates the acceleration collection at the first draw, and the blue color line indicates the second draw. When we are drawing the number "0" in the air without applying the Block Form (as it is shown in *Figure 5.1*), although the later part before it finishes starts to differ, the overlapping lines occupies most of the section. In contrast, in Figure 5.2 where the Block Form was applied, although the shape

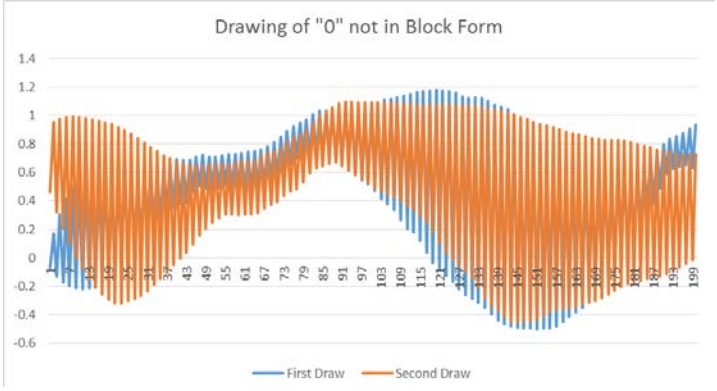

Figure 5.1. The data collection did not use block form

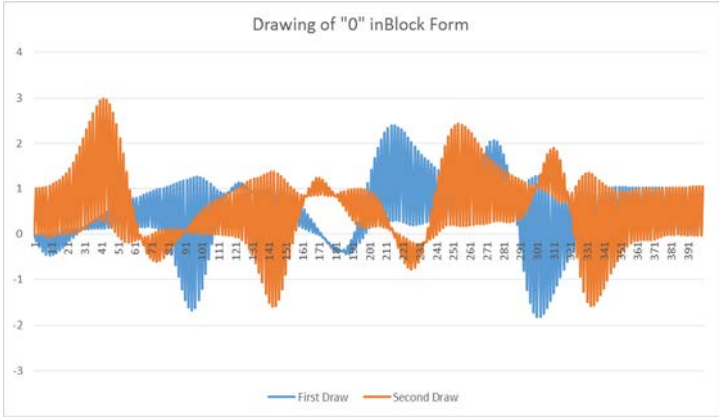

Figure 5.2. The data collection used block form

of acceleration lines of two individual drawings are almost identical, the offset resulted by the time different is huge. Moreover, because of such clear indicating of acceleration, any slight delay (e.g. 0.3s) of drawing will result in a huge difference of overall drawing. In addition, it is quite hard to control the 0.3s difference for a human being. Thus, although the logic and theory seems promising, such method of drawing has to be abandoned. This trial taught us the following facts:

- *Although less acceleration results in similar drawing curve for different numbers, it actually helps the data collected to be more delay tolerant*
- *Bigger acceleration do helps highlighting the features of individual number, but a more strict method of drawing has to be carried out so that to minimized the drawing delay*
- *One has to find the balance between big acceleration (better feature highlighting), and time delay tolerance.*

This is where we introduce the table drawing method.

IV. Table Drawing & Big Drawing

To reach the balance between the acceleration and time tolerance, we finally carried out the Table Drawing and Big Drawing method. The idea of this way is much more straightforward:

- *The numbers will be draw on the table instead of in the air this time. This is to increase the control over the drawing of each number so that to reduce the shakiness and time delay resulted by human hands*
- *The drawing of each number will still be the way indicated in Figure 3 (the smooth way). However, the drawing scale (size) is increase so that the acceleration can be increased, and the smoothness of each number is also maintained*

After this way is carried out, the testing result suddenly becomes accurate:

Trial	0	1	2	3	4	5	6	7	8	9
Success out of 10	8	6	7	7	10	9	8	10	9	10

As same as before, the mean square error was waited until it reduced to 0.13. As for the testing result, besides 1, 2 and 3, the rest of the numbers all reaches an identification rate of over 80%. The problem still rises for the drawing of 1. This is because of the "long" time, since the drawing 1 costs the least amount of time, and it usually ends before 2 seconds. This makes any delay of drawing having a relatively significant difference with the initial data collection pattern. Other than that, the testing result of this trial has reached our expectation.

V. Conclusion & Improves

After the successful implementation of LabView files, the main focus of this project shifted to the analyzation of the data collection, so that a more time tolerant and accurate weights matrices and be generated. By testing as well as summarizing the advantages/disadvantages of different approaches we had, we were able to finally combine the pros in the methods we tested and reduce the cons to a relatively acceptable degree. If the time is not a constraint for this project, some more improvements for the Block Form Drawing could be carried out. For example, if we can successfully find the way to regulate both of drawing shape and drawing time point, then an overall 90% recognition rate may not be impossible.