Deep Learning Network must be Discrete

A Preprint

Shoujun Zhou*

February 8, 2023

Abstract

I show in this paper that the information system of a computer and the cognition of human must be discrete, which is equivalent to a system based on integers. There are two mainly arguments to support this: 1) we can't construct a real number system with finite integers and 2) the system of real numbers is logically not measurable. Besides, I believe that the integer is more suitable for deep learning networks as they are originally discrete. To prove this, I conduct a series of experiments on the deep reinforcement learning.

Keywords Deep Learning Network · Discrete System · Measurement

1 Introduction

Everything is start from that I want do some discrete reinforcement learning, for it can be simple to analysis. Bute later I find it there should be no different between discrete and so-call continus.

The following article has 4 parts. The 2en paragraph introduces some basic concepts, including the description of that the learning network should be discrete, a general form of the full connected learning network. The 3rd paragraph presents a process for constructing a learning network of XOR manually. The 4th paragraph will discuss some ideas come out during the process in 4rd paragraph, such as the network is learning based on local linearity. The last part will make a conclusion.

2 Theory

2.1 The Cognition must be Discrete

I want to show two Phenomenons firstly.

The first one is that we can't construct a real number system from a discrete number system when we don't have infinite integers. The book PRINCIPLES OF MATHEMATICAL ANALYSIS wrote by Rudin Rudin [1986] shows a process of constructing the set of real numbers R from the set of real number Q. (And Q is equivalent to Z.) In that process, we need infinite subsets of Q to construct a real number, which means we need infinite elements of Q to construct a real number. So we can say that we need infinite integers to construct a real number. But in the real world, there is no infinite integers in a computer and (may) in our brains. We can't construct a true real number from a finite discrete system, not to say a information system of true real numbers.

On the other hand, let's suppose that there is a bit containing a real number. How can we know what it is exactly? One way is measuring it with a exact precision, but it appears nothing beyond a discrete finite system. With a exact precision means we don't know what it real is at the same time. Besides, how can a such bit be copied exactly? They have infinite precision, there seems no logic imaginable way to copy a real bit q1 to another real bit q2, or check if q1=q2. So, if such a information system of real number exists, it even can't be measurable. When we measure it with exactly precision, it downgrade to a discrete system. When we don't measure it, we will never know what it is.

*I am a Bachelor of Information Security in NUAA, and plan to learn Mathematics in ETH Zurich. GitHub address is https://github.com/reisenmiao and the contact email is zhoushoujun2021de@163.com.

It is no doubt that the information system of a computer is finite discrete which is equivalent to a system based on integers.

2.2 A General Form of the Deep Learning Network

A general form of the deep learning network is defined in the integer space Z as follows:

- There are N_x possible values of the input x which are denoted as $x_i = i, i = \{0, 1, 2...N_x - 1\}$;

- There are N_y possible values of the output $y \in \{0, 1, 2...N_y - 1\}$;

- The target function is $F(x) = y_x$, where y_x is random number of $\{0, 1, 2...N_y - 1\}$;

- The learning network is $f(x) = W(Max\{0, Ix + B\})$, where $W, I, B \in Z^{N_x - 1}$;

- $B = [b_0, b_1, b_1...b_{N_x-2}], b_i = -i$;

- $W = [w_0, w_1, w_2...w_{N_x-2}]$, $w_0 = y_1$, $w_1 = y_2 - 2w_0$, $w_2 = y_3 - 2w_1 - 3w_0$ ... $w_{N_x-2} = y_{N_x-1} - 2w_{N_x-3} - 3w_{N_x-4}... - (N_x - 1)w_0$

3 To Form a Deep Learning Network of XOR Manually

In this section I want to show the process of constructing the deep learning networks of XOR manually. XOR is a simple and classic example. The network is talked about in the book deep learning. But it only shows the configuration of the networks. There is no description about how to configure it manually. However, I think it is important to configure some deep learning networks manually to understeand how the deep learning networks works. So I'd like to show the following two processes of consturcting the networks of XOR. One is based on the thought of the general form mentioned above and another one is the network mentioned in the book deep learning.

3.1 To The General Form

For the XOR function, there are 4 possible inputs. So we set $x_i = i, i = \{0, 1, 2, 3\}$, where $f(0) = 0$, $f(1) = 1$, $f(2) = 1$ and $f(3) = 0$.

Then we can see, the graph of XOR is consisted by 3 line segment. So the middle layer should have 3 nodes.

Each node take a responsibility one local linear structure.

After configuring them according to the general form mentioned above, you can check it freely if it fits the XOR function totally.

3.2 The Original Input Form

For the original input, we can see there are two linearity structure in the graph. So there should be two nodes in the middle layer.

And the network which given by the book works as follows: From x to mid layer, it build a surface random, then change one of a point. From mid layer to y, it is a linear surface.

To a sumary, we can say that the network simulate a funtion based on the local linarity of the function.

References

W. Rudin. *Principles of Mathematical Analysis*. McGraw - Hill Book C., 1986. URL `https://books.google.de/books?id=frdNAQAACAAJ`.