FINAL REPORT

INT404: Artificial Intelligence

Topic: Simple Neural Network

Submitted by: Arijeeth Dash (11802165)

Roll Number: 07

Section: K18GX

Contents

Introduction

Neuron and its connection are part of a human brain and this is used as the basis to create AI which can think like human using the similar pattern as human brain do to perceive information or train themselves to understand a subject or do work on it.

Neural network is similar to human neuron connection and design like neuron with almost all working like it. But instead of using chemical energy or attaching emotional values the machine use different approach as it cannot produce the same way as human brain do. It uses maths and logic to understand or to solve a given problem. It requires training the machine with some datasets like practice to human beings.

Geoffrey Everest Hinton (born 6 December 1947) is an English Canadian cognitive psychologist and computer scientist, most noted for his work on artificial neural networks. Since 2013 he divides his time working for Google and the University of Toronto. In 2017, he cofounded and became the Chief Scientific Advisor of the Vector Institute in Toronto.

Abstract

The project undertaken is representation of basic and simple neural network, in this project the logic and methods were discovered back in 80's and still used till this date.

The project is demonstrated on python 3 and only NUMPY library is used, this library is used to solve complex mathematics which requires matrix and calculation regarding it. This project works on sigmoid graph and its properties of synoptics; this is similar to human brain which is trained by repeated action and this action is now taken by machine to learn and improve itself with given data.

This project predicts the values in one's or zero's according to value provided by user, it used sigmoid logic where the mean is 0 and repeated trained model of synaptic values.

Multiple function in the program help the overall project to work and to improve, this project works on certain synaptic values or weight and train itself multiple times before predicting the values given by user. This basic version of neural network is only demonstration and may not be useful in today's requirement since more advance and complex algorithm and logics are better and efficient.

Implementation

(Code)

```
import numpy as np


class SNN():


    def __init__(self):

        np.random.seed(1)

        self.sy_value = 2 * np.random.random((3, 1)) - 1 #value from -1 to 1 with mean 0


    def sigm_funct(self, x): #takes values and adds it and defines in range of 1 or 0

        return 1 / (1 + np.exp(-x))


    def sigm_val(self, x): #to calculate correct synaptic values

        return x * (1 - x)


    def model(self, in_value, ou_value, model_inc): #training function with values and errors produced

        for inc in range(model_inc):

            output = self.load(in_value) #modelling values passed through nn

            error = ou_value - output #rate of error
```

```
            changes = np.dot(in_value.T, error * self.sigm_val(output)) #error*input values*gradient of SF
            self.sy_value += changes #adjusting synaptic values

    def load(self, inputs): #output of nn
        inputs = inputs.astype(float)
        output = self.sigm_funct(np.dot(inputs, self.sy_value))
        return output


if __name__ == "__main__":

    nn = SNN() # starting nn

    print("synaptic values = ")
    print(nn.sy_value)
    #training values and output
    in_value = np.array([[0,0,1],[0,1,0],[1,0,0],[1,1,1],[1,0,1],[0,1,1]])
    ou_value = np.array([[0,0,1,1,1,0]]).T
    nn.model(in_value, ou_value, 10000)#Model the neural n/w

    print("Synaptic values after modelling : ")
    print(nn.sy_value)
```

```
#inputs
M = str(input("1st input 1/0: "))
N = str(input("2nd input 1/0: "))
O = str(input("3rd input 1/0: "))

print("Input Values = ", M, N, O)
print("Output Value = ")
print(nn.load(np.array([M, N, O])))
```

(Output)

```
synaptic values =
[[-0.16595599]
 [ 0.44064899]
 [-0.99977125]]
Synaptic values after modelling :
[[12.8002124 ]
 [-4.21782875]
 [-4.21771111]]
1st input 1/0: 0
2nd input 1/0: 1
3rd input 1/0: 0
Input Values =  0 1 0
Output Value =
[0.01451675]
>>>
```

Future Scope

This project is useful to understand how the machine thinks and operates; the algorithm used is not complex and hence cannot be used at various fields. The project demonstrates how with simple maths and logic, working of human brain can be shown via machine.

The other algorithm used in modern times can be used in various sector i.e. astronomy, medicine, climate ant etc.

Reference

1. https://en.wikipedia.org/wiki/Artificial_neural_network
2. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm
3. https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf
4. http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html