Replication of ARTIFICAL NEURAL NETWORK in C

First Semester Project made by:

Rayyan Minhaj	(20K-0143)
Kainat Afzal	(20K-0281)
Nashit Budhwani	(20K-0274)
Habeel Amin	(20K-0223)
Sabeerah Ahmed	(20K-0362)
Fabiha Atique	(20K-0369)

INTRODUCTION (AIM)

With our Project we aimed to replicate an Artificial Neural Network in C programming language, usually associated with the topic of AI. Traditionally it is used for machine learning by processing fed information and providing a suitable outcome (similar to how a human brain functions). It requires complex working done in layers using matrices (in our case).Similarly, the main working occurs in layers (hidden, input, output etc.). In our project we have omitted the backwards propagation as it is beyond our current educational standard. We have tried to feed information, specifically the attributes of a flower like its sepal length and with, petal length and width to determine the class of flower it belongs to.

BACKGROUND AND RESEARCH

In our research for this topic we indulged in help from our Programming Fundamentals professor Sir Musawar Ali who guided us through most of our queries along with help from numerous videos online and articles such as the following below. Also we have researched the multiplication of matrices and the working that takes place between and in our hidden layers.

https://www.coursera.org/lecture/visual-perception-self-driving-cars/lesson-1-feed-forward-neural-networks-87w0G

https://www.youtube.com/watch?v=qWK7yW8oS0I

https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6

PROJECT SPECIFICATIONS

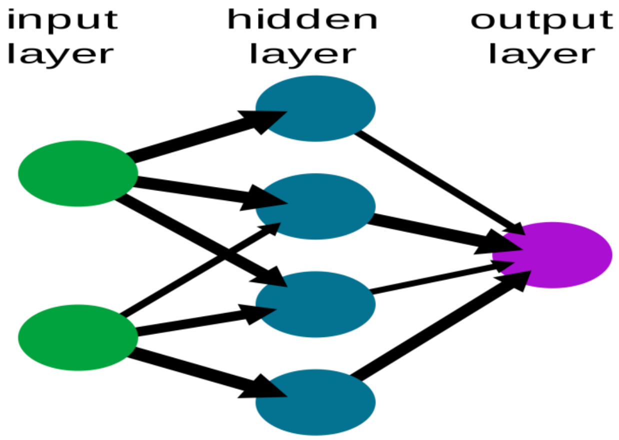
We have used several functions in our code more specifically for the generation of randomized matrix by judging the rows and columns of previously generated matrix (upto user input). Along with those there are separate functions for the multiplication itself, for the application of sigmoid, for the display of hidden weights and finally the application of argmax function. We are taking input from file provided to us and instead of user inputted values for first matrix layer we are taking values from the text file.

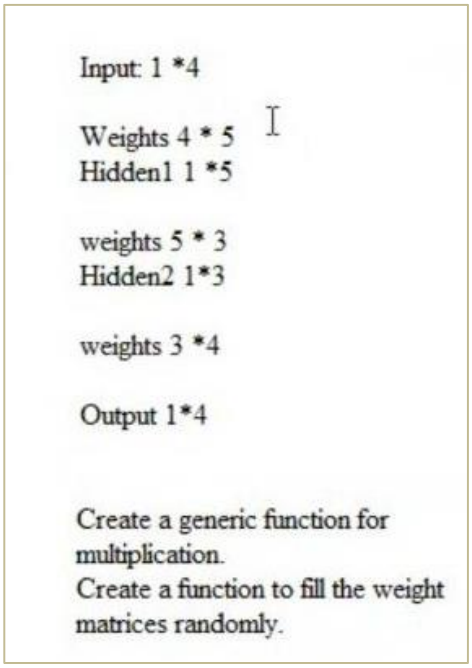
PROBLEM ANALYSIS

We did not face many problems, those which occurred were solved rather quickly from help of our sir and quick action in implementing the solution, such as: the issue revolving around the input of a single line of float values separated by spaces to store them in first matrix (whose rows and columns were user inputted). Similarly, matrix multiplication was something we achieved through trial and error as we had to develop a code which would multiply weights with random generated matrices whom rows and columns were decided through code.

SOLUTION DESIGN

Our plan of development/solution was to replicate the complex layering and multiplication that takes place between the hidden layers such as the sigmoid function and weight generation and storing them in matrices. We also followed the following pattern.




Input: 1 *4

Weights 4 * 5
Hidden1 1 *5

weights 5 * 3
Hidden2 1*3

weights 3 *4

Output 1*4

Create a generic function for multiplication.
Create a function to fill the weight matrices randomly.

IMPLEMENTATION AND TESTING

Our project is implemented on the data provided through our Sir, which are, as aforementioned, the features of a flower (petal length, petal width, sepal length, sepal width). With momentary and regular checking of our code by our Sir we were able to test the outcomes of our code through rigorous trial and error. A few examples of practical implementation of Artificial Neural Networks consists of determining numbers from handwritten text and making a prediction through the data set and achieving an accuracy score (argmax in our case).

TIMELINE OF ACHIVEMENT

Our Project was started roughly around the mid of November 2020 with constant updates and meetings we were able to utilize our time properly and focus on the project.

Our first task of many consisted of taking the input of number of rows and columns of our starting layer which was 1*4, due to there being 4 values in the data set. And matching them with the rows and columns of the random generated matrix filled with values ranging from 0-1 to make sure matrix multiplication was possible.

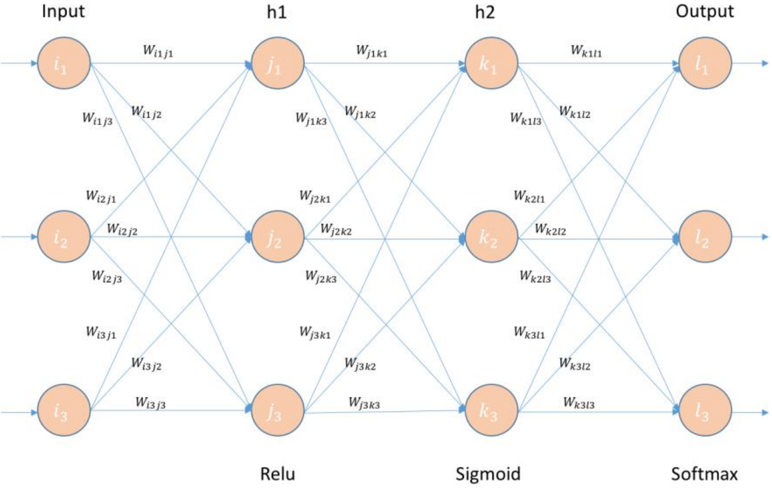
Through the help of each other we were able to create function for most of the following things to ease our working process and provide efficiency in our code:

- A randomizer function which generated random numbers between 0 and 1 and stored it in a matrix which was sent through the parameters.
- Sigmoid $S(x) = \frac{1}{1+e^{-x}}$ function which would apply the sigmoid equation to ensure our hidden layer matrix would **also** be in between 0 and 1.
- A matrix multiplier function which takes input of 2 matrices and checks if the col of first matrix is equal to row of the second and proceeds to create a 2d array (matrix) of the appropriate size and multiplies our parameter matrices and store in new one.
- Display function whose main purpose was just to take a matrix of any size and just display (print) on console.
- Argmax function which in integral part of a neural network to determine the largest value from the final matrix of 1*3 to determine the IRIS type.

And through the development of these functions we were able to create the first hidden layer of 1*5 size. Then we multiplied it by a random generated weight of size 5*3 to get a hidden layer

2 of size 1*3 followed by the final weight of size 3*3 which produced our solution or final output of size 1*3. Finally, our last step was to implement the values from file into first 1*4 matrix to get the argmax values.

RESULTS



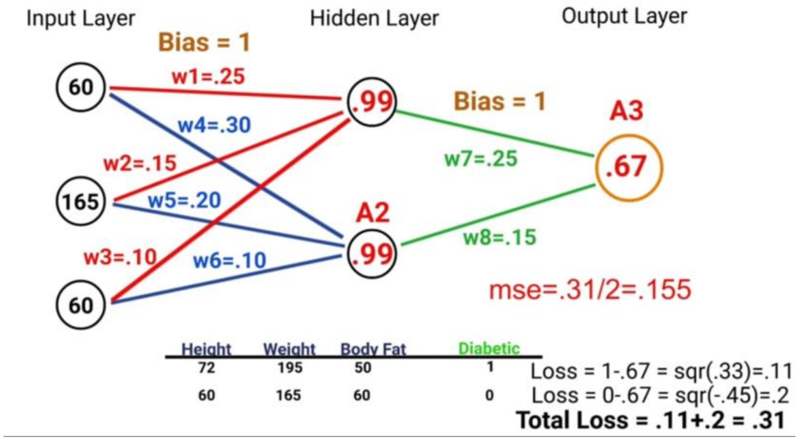
We were able to achieve a 1 by 3 matrix through calculation from which after application of the argmax were able to determine the position of the maximum value which determined the IRIS because we were provided the data set of 3 flowers. Similarly, in artificial neural networks through backwards propagation, the accuracy of the result can be determined but due to our omission of the backwards propagation we were merely able to achieve the position of the maximum value without much accuracy.

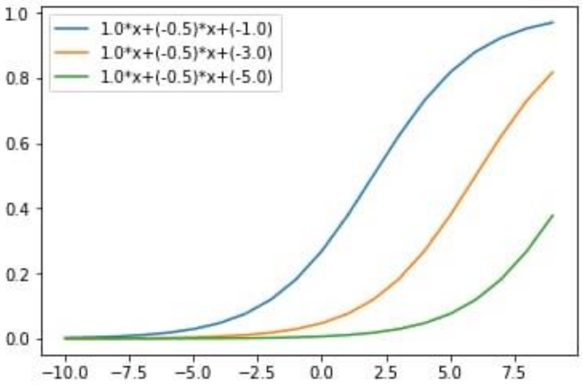
EXTRA WORK

For some additional extra work we have tried adding some functions to make our ANN feel and look more realistic such as:

- We have added a menu option for easy accessibility to train the ANN, view record of previously ran ANN, see the error and view the IRIS data set.
- We have added bias function. Biases tend to act as thresholds for determining which neurons are active and which are not by adding the user inputted value of bias and then passing it through an activation function (sigmoid in our case). In our implementation we have reduced the working to simply just adding the value of bias to our neurons (since we are not deciding which are active).
- Loss function to calculate the error. Error is considered the difference/discrepancy between achieved output and original data set. 0.0 is considered a perfect score and we have implemented MEAN SQUARE ERROR formula in achieving value of error. $MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - Y)^2$

Below the diagram explains inclusion of bias and loss. And a graph displaying the threshold properties of bias (right/left shift).





CONCLUSION

In conclusion, us students as a team learned a lot about the workings of Neural Network and explored the CS side of AI which we has developed in us an interest in the methods of Deep Learning, its amazing practical implementations and the leaps in technology that have been made through it. We would like to thank our Sir for providing us the opportunity to explore something new and interesting for our semester project. The neural network side is something most us would like to pursue in future anyways due to its increasing scope in the world and this has provided for us a stepping stone and notable mention for our career records.

THANK YOU