UC Riverside

Final Project Report

**Implementing a Neural Network**

**in**

**CUDA C**

Report by,

Rajat M. Jain

rjain032@ucr.edu

December 3rd, 2021

Aim: To implement a single hidden layer perceptron in CUDA C to design a binary classifier.

1. Overview:

In this project, I parallelize the serial approach of implementing Neural Networks by breaking down the steps into fundamental matrix operations.

1. Technical Description:

Implementing a neural network on the very low level is just operating functions like multiplication and subtraction on matrices. Taking my understanding from Lab 3 I define several kernels which perform different kinds of matrix operations like multiplication, dot product, subtraction and even performing the Sigmoid activation function for binary classification. Since it is a single hidden layer perceptron my code even facilitates backpropagation and updating of weights. In my approach I do not use any biases.

I operate on a few samples of data from the Iris dataset to implement a binary classifier on the basis of the sepal length, sepal width, petal length and the petal width. The binary classifier predicts 0 for “Setosa” species and 1 for “Versicolor” species.

The “main.cu” contains all the host side code while “nn.cu” contains all the device side code.

Functions and their description:

main(): Handles initializing all the parameters and calling the kernels.

dmatbymat(): Multiplies 2 matrices.

dmatsub(): Subtracts 2 matrices.

dsigmoid(): Applies the Sigmoid activation function for binary classification - f(x) = 1/(1 + e^-x).

ddersigmoid(): Applies the Sigmoid activation function for the derivative of the function - f'(x) = f(x)(1 - f(x).

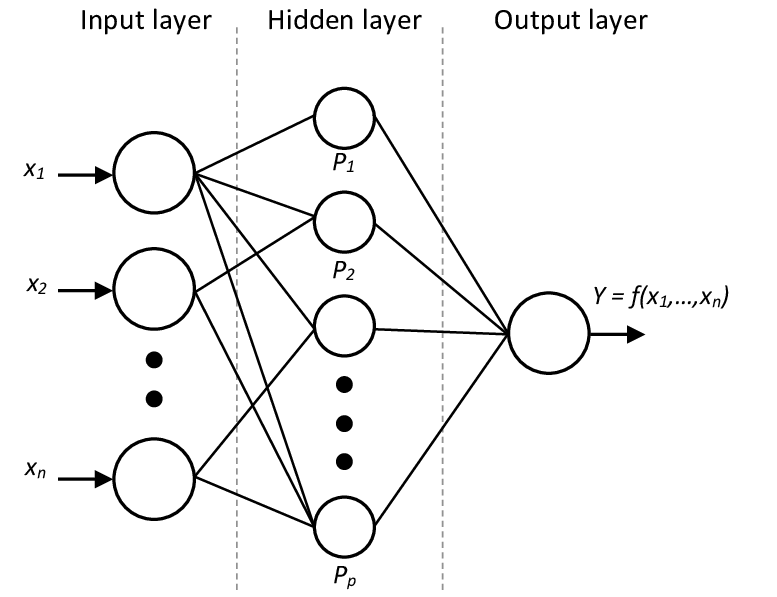
ddotprod(): Computes the dot product of 2 matrices.

dm1dotm2T(): Computes the dot product of 2 matrices

dm1Tdotm12(): Computes the dot product of 2 matrices

kdispmat(): Printing the input array dimensions (h,w)

kfit(): Running the algorithms for 50 epochs.



1. Status:
2. Technical Challenges:
3. Results:

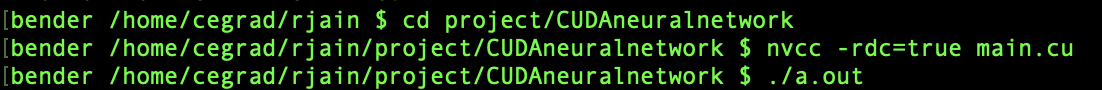
Text

Description automatically generated

1. Compiling:

Text

Description automatically generated



References:

1. <https://docs.nvidia.com/cuda/>
2. <https://stackoverflow.com/questions/27590166/how-to-compile-multiple-files-in-cuda>
3. <https://www.youtube.com/watch?v=gAgZkdTF4KQ&t=1s>
4. <https://developer.nvidia.com/blog/even-easier-introduction-cuda/>
5. https://github.com/UCR-CSEE217/lab3-matrixmultiply-rajatjain007