

Lab Practice - V

Assignment No. 04

Title :- Write a CUDA Program for :

1. Addition of two large vectors
2. Matrix multiplication using CUDA C

Objective :- Students should be able to perform CUDA Program for Addition of two vectors and Matrix Multiplication

Prerequisite :-

- Concept of CUDA
- Vector Addition
- How to execute program in CUDA environment

Contents for theory :-

- concept of CUDA, what is CUDA
- Addition of two vectors
- Matrix Multiplication

What is CUDA :

CUDA (Compute Unified Device Architecture) is a parallel computing platform and programming model developed by NVIDIA.

Using CUDA, developers can exploit the massive parallelism and high computational power of GPUs to accelerate computationally intensive tasks, such as matrix

Operations, image processing and deep learning.

Steps for Addition of two vectors using CUDA

1. Define the size of the vectors.
2. Allocate memory on the host. You can use the malloc function to allocate memory.
3. Initialize vectors. We can use loops to fill vectors with data.
4. Allocate memory on the device. You can use cuda function cudaMalloc to allocate memory.
5. Copy the input vectors from host to device.
6. Launch the CUDA kernel that will perform the addition operation.
7. Copy the result vector from device to host.
8. Free memory on the device.
9. Free memory on the host.

Steps for Matrix Multiplication using CUDA:

1. Matrix Initialization: The matrices are usually represented as 2D arrays.
2. Memory Allocation: You can use the standard C malloc function to allocate memory on the host.

3. Data Transfer : The third step is to transfer data between the host and the device for matrices.

4. Kernel Launch :- The fourth step is to launch CUDA kernel. Each thread in the kernel will compute one element of output matrix.

5. Device Synchronization : The fifth step is to synchronize the device to ensure that all kernel executions have completed before proceeding.

6. Data Retrieval : You can use the CUDA function `cudaMemcpy()` to transfer data from the device to the host.

7. Memory Deallocation : The C free funcⁿ can be used to Deallocate Memory on host and CUDA function `cudaFree()` to deallocate memory on the device.

Execution of program over CUDA Environment:

1. Install CUDA toolkit : we can download CUDA toolkit from the NVIDIA website and follow installation instructions.

2. setup CUDA environment : Once the CUDA Toolkit is installed set `PATH` and `LD_LIBRARY_PATH` environment variables to appropriate directories.

3. Write CUDA program: You need to write a CUDA program that performs the addition of 2 vectors and Matrix Multiplication.

4. Compile CUDA program: You can compile the program with nvcc compiler. The command to compile the program is:

```
nvcc -o program_name program_name.cu
```

5. This will generate an executable program. The command to run is:

```
./program_name
```

Conclusion: Thus we have executed Addition of vectors and Matrix Multiplication using CUDA.

Lab Practice - V

Assignment No. 05 [DL]

Title :- Linear regression using Deep Neural network: Implement Boston housing price prediction problem by Linear regression. Use Boston House Price predictionset.

Objective :- Students should be able to perform Linear regression using Deep Neural network on Boston House dataset.

Prerequisite :- Concept of linear Regression
Concept of Deep Neural network

Contents for Theory :-

1. Example of Linear Regression
2. How Deep Neural network works
3. Code explanation with output

Theory :-

What is Linear Regression :-

Linear regression is a statistical approach that is commonly used to model the relationship between a dependent variable and one or more independent variables.

It assumes a linear relationship between the variables and uses mathematical methods to estimate the coefficients that best fit data.

Example of linear Regression:

A suitable example of linear regression using deep neural network would be predicting the price of a house based on various features such as the size of the house, no. of bedrooms, the location and the age of the house.

During training, the weights and biases of the neural network would be adjusted to minimize the difference between the predicted price and actual price of house.

Concept of Deep Neural Network:-

A Deep neural network is a type of machine learning algorithm that is modeled after the structure and function of the human brain.

It consists of multiple layers of inter-connected nodes, or artificial node neurons that process data and learn from it to make predictions or classifications.

How Deep Neural network work:-

Boston House Price Prediction is a common example used to illustrate how a deep neural network can work for regression task. Here's how a deep neural network can work for Boston House price prediction:

1. Data preprocessing: The first step is to preprocess the data. This involves normalizing the input features to have a mean of 0 and Standard Deviation 1.

2. Model Architecture: The first layer is the input layer, which takes in the normalized features. This is followed by several hidden layers.

3. Model training: During training, the weights and biases of the nodes are adjusted based on the error between predicted output and the actual output.

4. Model Evaluation: Once the Model is trained, it is evaluated using testing set. The performance of the model is measured using metrics.

5. Model Prediction: Finally, the model can be used to make predictions on new data, such as predicting the price of a new house in Boston.

Conclusion: In this way we can Predict the Boston House Price Using Deep Neural Network.