Learning Mesh-Based Simulations for Predicting Weapon Effects

Program: AFRL Scholars

Mentor: Dr. Alla Kammerdiner, *NRC/AFRL, Ph.D.Industrial and Systems Engineering*

Researchers:

- **Alexander B. Vereen**, *Univeristy of South Carolina, B.S.E Mechanical Engineering*
- **Caleb I. Fryer**, *Texas A&M Univeristy, B.S Mechanical Engineering*

Learning Mesh-Based Simulations for Predicting Weapon Effects

Project Summary

Overview

This topic will address the fundamental challenges in real-time prediction of weapon effects (such as a tradeoff between precision and speed). Currently weaponeers rely on engineering models for lethality. These are low-fidelity models calibrated to a subset of munitions and environments. The resulting uncertainty often requires overallocation of munitions, which reduces the warfighter's capabilities against adversaries. High fidelity simulations can reduce uncertainty but are too computationally intensive and time consuming to be used by weaponeers. The proposed research will utilize recent advances in Machine Learning (ML) methods to rapidly provide surrogate data to weaponeers (based on offline training on high fidelity simulation data). Specifically, selected scholars will learn how to apply MeshGraphNets to learn and predict mesh-based finite element simulations. Also, the scholars will compare MeshGraphNets with other ML methods. The scholars will advance their ability to code and perform numerical analysis. This work will provide the scholars valuable experience with state-of-the-art code for ML and high-fidelity simulations.

Project Description

Introduction

Proposed Study

Propose the usage: the usage of mesh graph net as a replacement for simulation for analyzing operational deployment of ordinance Background information: Provide a brief overview of the project or problem that the study aims to address.
Scope of the study: Outline the boundaries and limitations of the proposed study to set expectations for the readers.
Importance of the study: Explain why this study is crucial for the project or the organization.

Current State Of The Technology

Existing technology overview: Provide an overview of the current technology, methodologies, or processes related to the subject matter.
Strengths and weaknesses: Identify the advantages and limitations of the current technology, highlighting areas that need improvement.
Industry trends: Briefly mention any emerging trends or advancements in the field that are relevant to the study.
Gap analysis: Identify the gaps between the existing technology and the desired goals to show the need for further investigation.

Intended Impact

Project objectives: Clearly state the intended outcomes and objectives of the proposed study.
Potential benefits: Explain the potential positive impacts the study could have on the project, organization, or industry.
Risk assessment: Address any potential risks or challenges that may arise during the study and how they will be mitigated.
Timeline and resources: Provide an overview of the estimated timeline and resources required to conduct the study.

Background

Nueral Networks

Graphs

Meshgraphnets

Environmental Configuration

Overview

PyTorch and TensorFlow are two of the most popular deep learning frameworks used for building and training machine learning models. When setting up a development environment for working with these frameworks, having a Linux native environment or WSL2 (Windows Subsystem for Linux) can significantly ease the configuration process and subsequent usage with GPU acceleration, provided available Nivdia hardware. You can confirm your system hardware supports CUDA and cuDNN here. Both PyTorch and TensorFlow offer GPU support, which allows for faster training times.

A Linux native environment or WSL2 is advantageous for configuring a machine learning environment in Python due to better package compatibility, efficient package managers, robust command-line tools, improved GPU support, enhanced reproducibility, and alignment with the machine learning community's practices.

Windows Subsystem For Linux

This section will breifly walk through how to install and enable WSL2 for your windows machine, documentation found here.

- Launch PowerShell or Windows Command Prompt as an administrator by right-clicking and choosing "Run as administrator." then enable the usage of WSL and virtual enviroments by entering

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform
```

- Next either download WSL from the window store or input the command

```
wsl --install
```

  and then reboot your machine to complete the installation process.

- After the restart, open the Microsoft Store and search "Linux." You'll find various Linux distributions available. Choose the one that supports your intended version of CUDA refrenced below.

- After the installation is finished, launch the distribution from the Start menu or by searching for its name. The first time You run the Linux distribution, you will be prompted to set up a new Linux user and password.

- Configure your default linux distrobution to the installed distrobution using the powershell command

```
wsl -s <DistributionName>
```

You may now execute commands from Powershell to run in your install linux distrobution using the command "wsl" or "bash" to enter linux terminal.

Linux Python Environment Management

Before installing PyEnv, found here, make sure you have the necessary dependencies in your WSL environment:

- From bash within wsl run these commands

```
sudo apt update
sudo apt install -y git curl make build-essential libssl-dev zlib1g-dev
libbz2-dev libreadline-dev libsqlite3-dev wget llvm libncurses5-dev xz-
utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev
```

- Next you may install PyEnv using curl

```
curl https://pyenv.run | bash
```

- Next you must add PyEnv to the Path environment variable using the following commands

```
export PYENV_ROOT="$HOME/.pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init --path)"
source ~/.bashrc
```

- You may now install and manage multiple python enviroments separate from your Linux system Python environment using the follow commands:
  - to list available Python versions

```
pyenv install --list
```

  - to list install a Python version

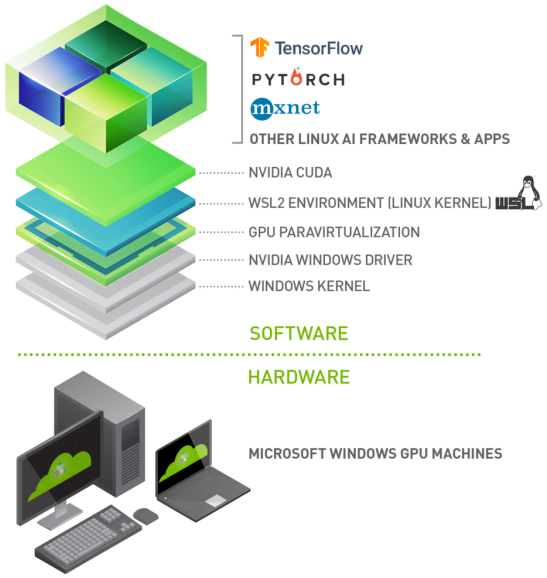
```
pyenv install <VERSION NUM>
```

  - to set the PyEnv virtual environment as the primary Python in enviroment

```
pyenv global <VERSION NUM>
```

GPU Configuration

Now supported in WSL2, available for Window 10 and 11, and the Linux OS natively CUDA and cuDNN for running machine learning applications such a TensorFlow and PyTorch. Following the guidance available from Nvidia here the current release WSL 2 v1.2.5, released April 20, 2023, supported by Windows 10 and 11, uses a paravirtualizationShown in Fig.4.4 of the GPU architecture for the Linux subsystem to interact directly with the Window GPU driver. Whether using WSL2 as outlines in this guide or native linux installations the drivers can be found available from Nvidia here to enable the usage of CUDA and cuDNN for machine learning applications.



Environmental Configurations

Detailed here is the environments used in this study, attached in the github :

TensorFlow V1.X

- *Python Version* == 3.6.x
- to install using pip

```
    pip install -r requirements.txt
```

- to set the PyEnv virtual environment as the primary Python in enviroment

```
    tensorflow-gpu==1.15
    dm-sonnet==1.36
```

PyTorch

- *Python Version* == 3.10.x
- to install using pip

```
    pip install -r requirements.txt
```

- to set the PyEnv virtual environment as the primary Python in enviroment

```
    tensorflow-gpu>=1.15,<2
    dm-sonnet<2
    matplotlib
    absl-py
    numpy
```

Code Base

Nueral Networks

Graphs

Meshgraphnets

Results

Metrics

Test Model: CFD

Test Model: Impact

References Cited