Project Proposal for Google Summer of Code 2023

Google Summer of Code

ML
4
SCI

End-to-End Deep Learning

Regression for Measurements with the CMS Experiment

Mentors

→ Ruchi Chudasama (University of Alabama)
→ Emanuele Usai (University of Alabama)
→ Shravan Chaudhari (New York University)
→ Michael Andrews (Carnegie Mellon University)
→ Eric Reinhardt (University of Alabama)
→ Samuel Campbell (University of Alabama)

Machine Learning for Science

TABLE OF CONTENTS:

CONTACT INFORMATION:

Name: Vishak K Bhat

University: Indian Institute of Technology (IIT), Dhanbad

Email: vishak.bhat5@gmail.com | 21je1047@iitism.ac.in

GitHub: vishak-github

Kaggle: vishak-kaggle (expert)

LinkedIn : vishak-linkedin

CV: Vishak-CV

Phone : (+91) 7760289129

Time zone: Indian Standard Time (UTC +05:30)

Location: Bengaluru, India.

PROJECT SYNOPSIS and PROBLEM DESCRIPTION:

The **Large Hadron Collider (**LHC**)** at **CERN** is one of the largest and most complex scientific instruments ever built. Its primary goal is to study the fundamental nature of matter and energy by colliding particles at extremely high energies. One of the key challenges of **analyzing data** from the **LHC** is the large amount of information produced by its detectors.

In particular, the **Compact Muon Solenoid (CMS) detector** records a vast array of data from particle collisions, including energy deposits, particle tracks, and other information. CMS acts as a giant, high-speed camera, taking 3D “photographs” of particle collisions from all directions up to 40 million times each second. The CMS collects a few **tens of Peta-Bytes** of data each year.

In order to extract meaningful physics information from this data, advanced data analysis techniques such as deep learning, Convolutional Neural Networks(CNN) are developed. The **aim** of this project is to develop and integrate an end-to-end deep learning regression model for estimating the properties of a **simulated top quark** pair event. The model will be trained using the above data and optimized to accurately predict the properties of the event. Additionally, the project aims to extend the currently **integrated E2E CMSSW** prototype to include the regression model inference, allowing for more efficient and effective analysis of top quark pair events.

PROJECT APPROACH and IMPLEMENTATION:

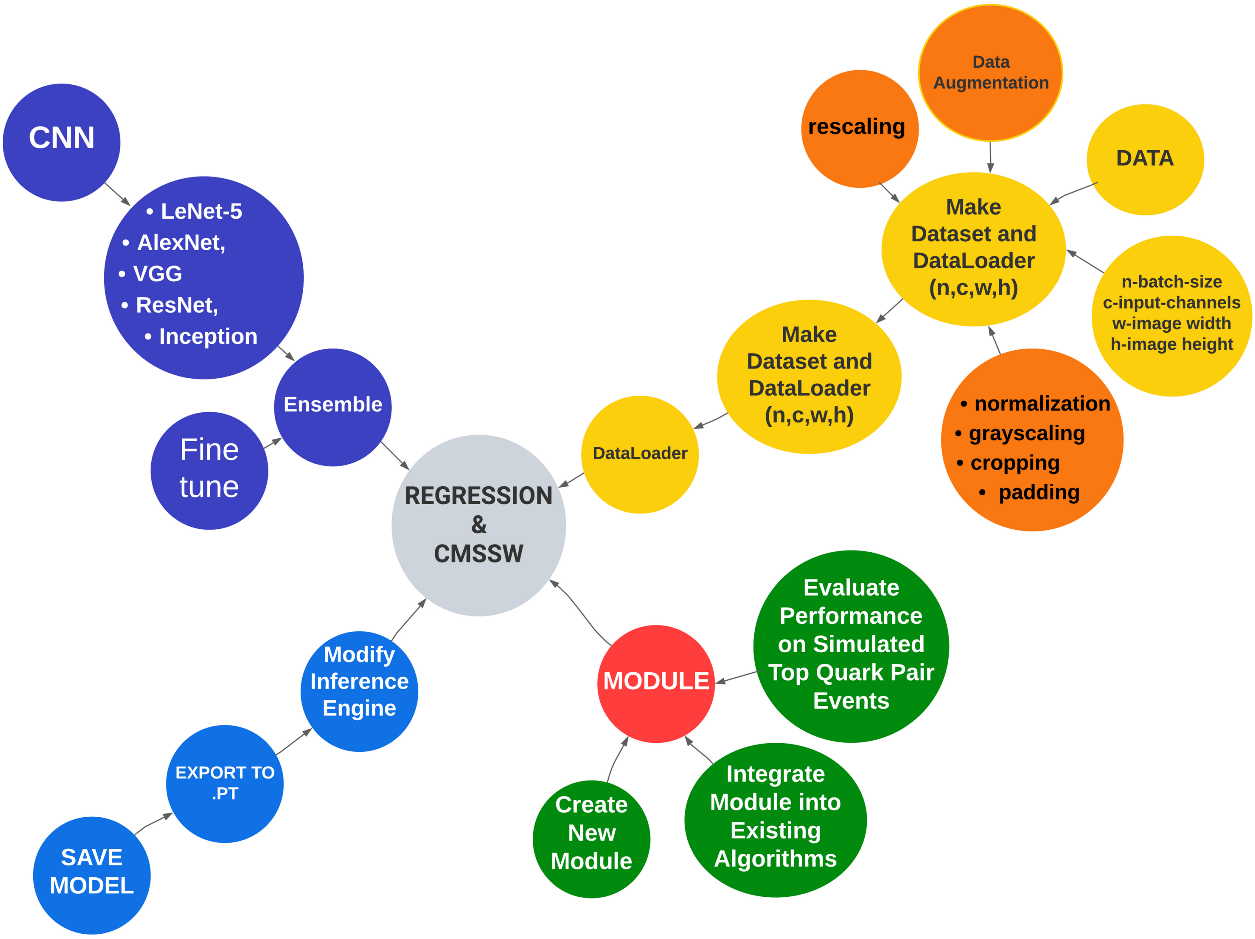

fig1: Mind map of the approach

DATA PREPROCESSING:

I will use various preprocessing techniques such as **rescaling**, **normalization**, **gray scaling**, **cropping, padding, data augmentation, feature extraction,** and **transfer learning** to prepare the image data for the regression model. I will then split the data into training, validation, and testing sets to evaluate the performance of the model.

I will use libraries such as **OpenCV, scikit-image, NumPy, Pytorch, TensorFlow,** and **Keras** to implement these preprocessing techniques. I will load and resize the images using **bilinear or nearest neighbor** interpolation, normalize the pixel values using NumPy operations, convert the images to grayscale using OpenCV , and apply **random rotations,** flips, and crops to generate new images using Pytorch.

I will also **extract** meaningful features such as **edges, textures, or corners** using filters such as **Sobel, Laplacian, or Canny,** which are available in OpenCV. Additionally, I will use pre-trained models such as **VGG or ResNet** to extract features from images and speed up training using **transfer learning**.

By performing these preprocessing techniques on the image data, I will ensure that the data is in a suitable format for the regression model and that it contains relevant features that can help the model make accurate predictions.

MODEL TRAINING:

I will train **five** different **CNN**(Convolutional Neural Network) architectures, including **LeNet-5, AlexNet, VGG, ResNet,** and **Inception**, on the same dataset using Pytorch. Each of these models will be trained independently, with their own hyperparameters and training strategies.

Once the models are trained, I will use their **outputs as features** and **feed** them into a final regression model. There are several ways to **ensemble** the models, including averaging the outputs, using a simple linear regression or another neural network, or using a more sophisticated technique such as a gradient boosting algorithm. To ensure that the ensemble is effective, I will evaluate the diversity of the individual models and adjust their hyperparameters as necessary to achieve optimal performance. This can involve **tuning** the learning rate, the number of layers, the size of the filters, or other parameters that affect the performance of the models.

CMSSW:

To extend my model to **CMS Software** (CMSSW), I will start by **saving** the trained regression model in a format that can be **easily loaded** into CMSSW. I will use Pytorch to export the trained model in .pt format.

Next, I will modify the **inference engine** of the CMSSW software to **include** the new regression model. This will involve **creating a new module** that can load the trained model and perform the necessary calculations to estimate the particle properties. I will also need to integrate the module into the existing reconstruction and identification algorithms of the CMS experiment.

Once the new module is integrated, I will **evaluate** the performance of the extended CMSSW prototype by running it on **simulated top quark pair events** and comparing the results to those obtained using the existing reconstruction and identification algorithms.

PROJECT DELIVERABLES:

1. A **trained** end-to-end deep learning regression **model** for estimating the properties of a simulated top quark pair event.
2. **Integration** of the trained regression model into the **CMS Software** (CMSSW) offline and high-level trigger systems, allowing for use in **reconstruction algorithms**.
3. A **modified inference engine** for the CMSSW software that **includes** the **new regression model.**
4. **Evaluation** of the performance of the **extended CMSSW** prototype on simulated top quark pair events.
5. **Documentation** of the end-to-end deep learning regression model and its integration into the CMSSW software.

EVALUATION TASKS:

I had completed the evaluation tasks mentioned in the stipulated deadline. The Jupyter Notebook has been commented adequately.

As mentioned, I have made a model to **classify electrons and photons** and got the minimum ROC AUC of 80%. This can be seen from the graph of ROC(TASK 1). Then I have made a model to classify the particles - **Quarks and Gluons**.(TASK 2). Lastly I made a **model for the regression problem** where the data was split into 20 percent test set in order to prevent overfitting. (TASK 3-(2))

All the models were trained in GPU.

The GitHub link of the evaluation task - **GitHub** .

PROJECT DETAILED TIMELINE:

PHASE/WEEK	DATE	WORK DESCRIPTION
COMMUNITY BONDING		
COMMUNITY BONDING	May 4- May 12	→ Familiarizing with the Community and Organization Standards

		→ Discussion of problems and final goals
	May 13- May 21	→ Define the project’s outcomes more clearly → Validate them with mentors.
	May 22- May 28	→ Break project goals into smaller, trackable issues for better analysis of progress and milestones. → Complete initial setup of working environment.
PHASE 1- May 29 - July 10		
Week 1	May 29 - June 4	→ Familiarize with the top quark pair event dataset. → Read relevant literature on the five different CNN architectures and their applications in image classification and regression tasks.
Week 2	June 5 - June 11	→ Preprocess the top quark pair event dataset and split it into training and validation sets. → Implement the two CNN architectures in Pytorch and train them independently on the training set.
Week 3	June 12 - June 18	→ Implement the other three CNN architectures and optimize them. → Tune the Hyper parameters of the models already made.

Week 4	June 19- June 25	→ Explore different Ensembling techniques, such as averaging the outputs or using a simple linear regression or another neural network. → Choose the most effective Ensembling technique based on the validation set results.
Week 5	June 26 - July 2	→ Train the final regression model using the outputs of the five CNN models as features. → Evaluate the performance of the model on the validation set and adjust hyperparameters as necessary.
Week 6	July 3 - July 10	→ Make all the code(notebook) written in presentable format for the mid evaluation → Making the summary and report of all the models
PHASE 1 Evaluation: July 10 - July 14		
PHASE 2: July 14 - August 21		
Week 7	July 14 - July 20	→ Export the trained regression model in .pt format using Pytorch. → Explore more about the inference engine of the CMSSW software
Week 8	July 20 - July 27	→ Modify the inference engine of the CMSSW software to include the new regression model.

		→ Create a new module that can load the trained model and perform the necessary calculations to estimate the particle properties.
Week 9	July 28 - Aug 3	→ Integrate the module into the existing reconstruction and identification algorithms of the CMS experiment. → Test the extended CMSSW prototype by running it on simulated top quark pair events and comparing the results to those obtained using the existing reconstruction and identification algorithms.
Week 10	Aug 3 - Aug 10	→ Summarizing the entire work done, tabulating and plotting the results obtained. → Completing the documentation, integrating all code.
Week 11	Aug 10 - Aug 21	→ Buffer week and further enhancements
Final Evaluation: Aug 21 - Aug 28		

About Me and Motivation for GSOC:

Overview:

I am a second year undergraduate pursuing Integrated MTech in Mathematics and Computing from the Indian Institute of Technology, Dhanbad(IIT ISM Dhanbad), India.

I am a typical geek who loves programming and enjoys problem-solving and making side projects as a part of hobby coding. Along with my friends, I manage a university-level open-source community, Cyber Labs, where we regularly participate in discussion of research papers related to computer vision and take part in machine learning competitions.

The pride in the feeling that my code will cause an impact in the lives of millions of people who will use it is unparalleled. Moreover, it allows me to grow as an individual and learn how to work in a team with such a big community. I have also been active in introducing people to the world of open source and also getting them involved with various open-source projects and communities.

SKILLS:

C++(STL, Armadillo, mlpack, Boost), Python(NumPy, Pandas, Matplotlib, Scikit-learn),MATLAB, Bash, SQL, Pytorch, TensorFlow(elementary proficiency).

EXPERIENCE:

- Completed the below **courses** from Coursera:
  - Convolutional Neural Network by Stanford, Coursera.
  - Improving Deep Neural Network by Stanford, Coursera.
  - Machine Learning by Stanford, Coursera.
  - Neural Network and Deep Learning by Stanford, Coursera.

Under **CyberLabs** made few projects like:

- "Emotional fool"- Which basically detects the emotions of the human using open CV. Further it displays the emoji of the reaction detected.
- "Dementia Classifier"- Given the image of the CDT(Clock Drawing Test), this model classifies the severity of the disease ranging from 0-5.

Kaggle: participated in many Kaggle competitions and currently **Expert tier** in Kaggle.

Represented college in the inter IIT tech fest for the machine learning problem statement.

REFERENCES:

1. ML4SCI Project Ideas
2. Vision Transformers for End-to-End Particle Reconstruction for the CMS Experiment
3. Evaluation test
4. End-to-End Physics Event Classification with CMS Open Data
5. End-to-End Jet Classification of Quarks and Gluons with the CMS Open Data
6. GSoC 2023 Projects Related To CMS
7. CMS Experiment
8. Large Hadron Collider
9. CMSSW
10. CMS experiments
11. Ensemble
12. LeNet-5

13. AlexNet
14. VGG
15. ResNet
16. Inception

THANK YOU