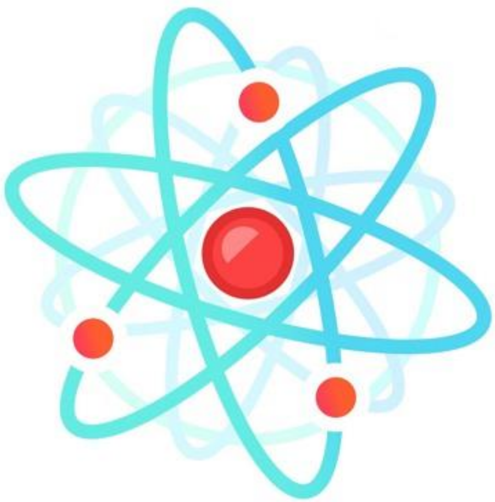

Implementation of Deep Learning Compression Algorithms for Muon Momentum Estimation in Compact Muon Solenoid Trigger System

Compact Muon Solenoid or CMS is a system developed for detecting Muons. Muons are charged, heavy elementary particles in atoms. Since Muons are a particle with a relatively large mass, traveling close to the speed of light, they have a high penetrative power. As such, it’s imperative to study about Muon Momentum. However, Muon Momentum Estimation is not feasible [1]. This project proposes to conduct a case study on the implementation of deep learning compression algorithms for Muon Momentum Estimation in the CMS Trigger System on the Level-1 trigger. This project also will be implementing a emulator for the level-1 trigger system.

Mohamed Ayoob Nazeem

Table of Contents

About Me

I am Ayoob, a student studying B.Eng (Hons) Software Engineering from the University of Westminster, UK. I am doing the degree in IIT, Sri Lanka. I also **interned in the Big Data and Data Science Team** of Zone24X7 for a year, and now in my final year of my degree. As a Big Data Science intern, my enthusiasm lies in impelling vast amounts of data to yield structure and harmony from within. I received a **scholarship to do a nano-degree in deep learning** by Facebook Developer circles at Udacity.

As an intern at Zone24X7 I have engineered machine learning solutions primarily in Python, PyTorch, Apache Spark, and Scala, I have also dipped my toes in Bash Scripting, Gradle builds, Groovy and am familiar with the CICD tools (Jenkins and Tonomi). I'm comfortable with languages such as Java, and C++ through exposure at my campus.

Regarding open source projects, I have participated and mentored my junior students to complete the Mozilla Global Sprint. I also co-founded the AI Research Club on my campus to carry out AI outreach programs. I write articles about Machine Learning in Medium as well. I **mentored interns on a NASA star classification project**. Mentoring women in tech on Astronomy and Computer Science. The project can be found as a GitHub repo here NASA-Star-Classification.

I also did an **internship** under Dr.Rukshan Batuwita, a Ph.D. in the University of Oxford on deep learning and research methodology. I have done many more courses on machine learning and data science. A comprehensive list can be found in my LinkedIn profile in the contact information below.

My interests lie in Applied Artificial Intelligence and Machine Learning. I am currently writing my thesis on Hyphersphere optimization for image synthesis for the Final year project for my degree. I hope to do my higher studies in evolutionary models and social automata theory. When I'm not reading books on AI or astronomy, I can be found hiking or riding my bicycle. Sometimes I bake stuff. Mathematics and Computer Science is my dope. I am a natural Pythonista.

A. Contact Information

Student Name: Mohamed Ayoob Nazeem
Mobile Phone: +94765452123

Email: mohamedayoob01@gmail.com (Primary) , nazeem.2016343@iit.ac.lk (Institute) , w1654551@my.westminster.ac.uk (Institute)

LinkedIn: mohamedayoob7
Skype: mohamedayoob01@gmail.com

B. University Referee

University: Informatics Institute of Technology, Sri Lanka (IIT Sri Lanka)
Degree program: Bachelors in Engineering (Hons) Software Engineering
Progress: 4th year of 4 years

Contact to Verify:

Guhanathan Poravi - Senior Lecturer, Informatics Institute of Technology.
Informatics Institute of Technology, 57 Ramakrishna Road, Colombo 06, Sri Lanka.
Tel: +94 112 360212 / 402(Ext)
Mob: +94 768 209 744
Email: guhanathan.p@iit.ac.lk

C. Internship Referee

Company: Zone24X7
Title: Big Data and Data Science Intern
Progress: Internship Completed

Contact to Verify:

Hansa Perera – Associate Architect – Data Science and Machine Learning
Zone24x7 (Private) Limited, 460, Nawala Road, Koswatta, Sri Lanka
Tel: +94 11 2033900 / 141 (Ext)
Mob: +94 76 278 8768
Email: hansap@zone24x7.com

1. Background to the problem

1.1. Prologue

Physics has always been the subject of finding the dynamics of the unknown. Or in other words, investigating the changing nature of reality and modeling them mathematically in an attempt to predict nature. Physics has many branches of study. One of the newest spheres of interest in physics is particle physics. Interest in particle physics started in the late 19th century. Pioneered by physicists like John Dalton and Murray Gell-Mann.

1.2. Muons

Particle physics, investigate the structure and composition of atoms and atomic particles such as quarks, mesons, baryons, etc. All of these elementary particles are classified and explained by the standard model of physics.

Muon is a particle that was discovered in 1936 by physicists Carl Anderson and Seth Neddermeyer. It is classified as a lepton in the standard model. A muon is extremely unstable, as it decays rather quickly into an electron and a pair of neutrinos. They have high penetrative power and due to the decay, they have high ionizing power as well. Muons are considered heavy particles when compared with electrons and neutrinos. Muons are used in radiography and tomography.

1.3. Compact Muon Solenoid

Detecting Muons is one of the principal tasks of Compact Muon Solenoid or CMS. Muons are important in finding the god particle Higgs Boson. One of the dominant decays of Higgs boson is into 4 Muons [2]. Since Muons have high penetrative power, they are likely to be detected easily by the CMS. This is one of the main reasons CMS was built. However, the vast stores of data accumulated still is a treasure trove of hidden knowledge for physicists.

1.4. CMS Trigger

When CMS is smashing particles at its peak there are billions of proton-proton collisions every second. Here the data ingress is of very high veracity and velocity. There is simply too much data in such events. Most of the data are explainable facts and previously observed phenomenon. We, therefore, need a “Trigger” that can select potentially interesting and anomalous events that can be consolidated on a computer for further analysis.

Level 1 trigger or hardware online trigger looks for simple criterions on the events. For instance, particles with large amounts of energy or unusual combinations and consolidate such data in a server farm of more than 1000 computers. The overall architecture of an upgraded trigger system is comprehended as well [3].

2. Requirement Survey

2.1. Problem definition

CMS has triggers to isolate anomalous events happening in the particle accelerator. Machine learning algorithms are being used Level-1 trigger (hardware trigger) to estimate the momentum of the Muons traveling through the accelerator. The current algorithm in usage is a boosted decision tree for momentum regression. However, CMS is studying on utilizing deep learning algorithms at the Level-1 trigger.

$$muon\ momentum = mass\ of\ muon * velocity\ its\ travelling$$

As the CMS ingresses thousands of data points per second, hence the Endcap Muon Track Finder or EMTF (which is part of the Level-1 trigger) has only about 500 nanoseconds to make the inference before the data is flushed. Such a critical mission dictates that the model must have the following properties.

a. Highly optimized model inference
b. High accuracy
c. Low latency in inference
d. High precision and recall

Thus, we can conclude the project problem definition. Next, we will be having a look at the rich picture of the project.

2.2. Rich Picture

Figure 1 below illustrates the rich picture of the project and identifies the scope of the work in the project. Based on a cursory reading we can identify the data that CMS collects is being massively down sampled. This has other potentials for autoencoders to reconstruct the data as well.

Figure 2 below illustrates the scope of the project and the components of the Level-1 trigger systems. We also narrow down on the mission area where the project improvements are to be carried out.

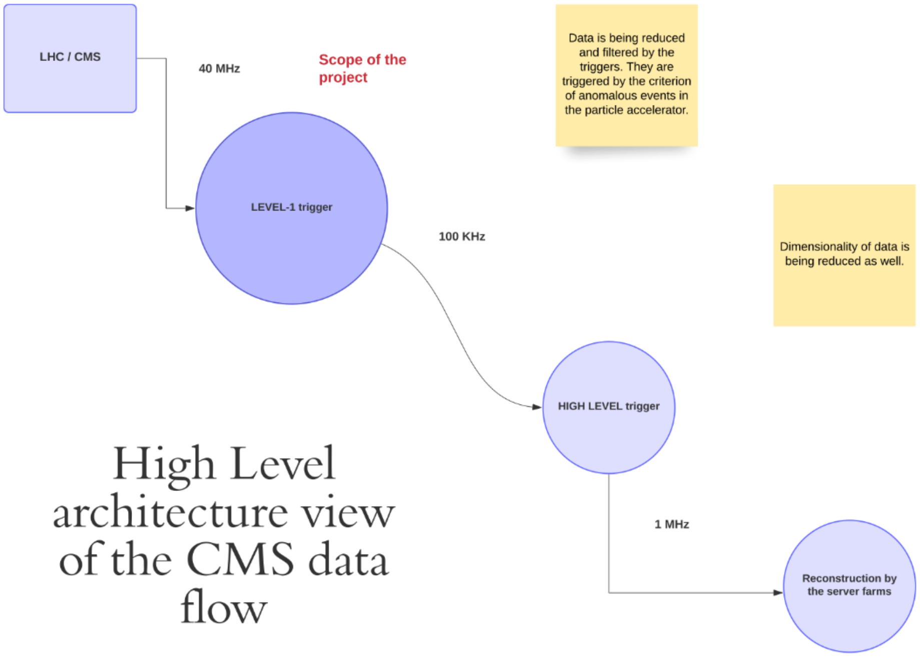
LHC / CMS
40 MHz
Scope of the project
Data is being reduced and filtered by the triggers. They are triggered by the criterion of anomalous events in the particle accelerator.
LEVEL-1 trigger
100 KHz
Dimensionality of data is being reduced as well.
HIGH LEVEL trigger
High Level architecture view of the CMS data flow
1 MHz
Reconstruction by the server farms

Figure 1- High Level Scope of the project

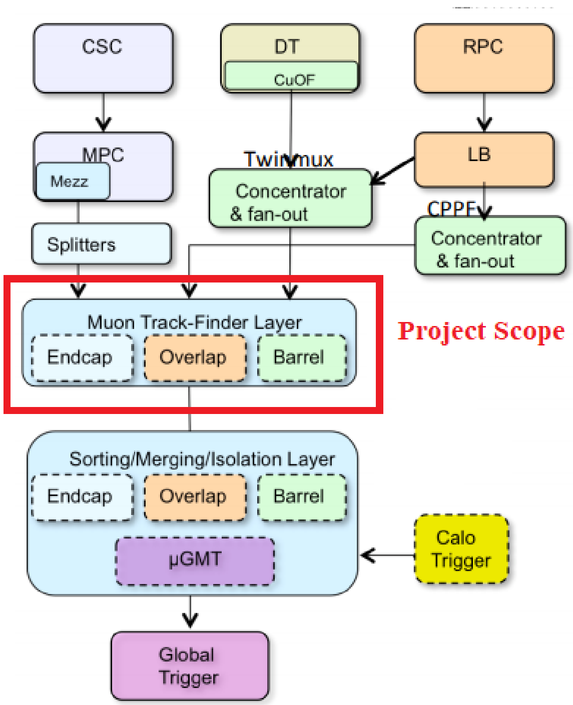

Figure 2 - Project scope narrowed down [4] [5] (emphasis added) Level-1 Trigger

2.3. What can be reused?

The dataset ingress software infrastructure can be reused. As we will be implementing and benchmarking algorithms for momentum regression based on the collected dataset for different algorithms.

2.4. What needs to be replaced?

The boosted decision tree regressor in the EMTF should be replaced with a preferably a neural network model that has the properties mentioned in the problem definition.

2.5. Required skills

a. Experience in machine learning with machine learning frameworks such as PyTorch and Tensorflow.
b. Experience in Python and C++.
c. Experience in working with codebases and version control systems.
d. Working with systems close to the hardware.
e. Seminal knowledge of the CMS and trigger systems.
f. Working knowledge of particle physics.
g. Experience with Linux.

3. Project Design and Literature Survey

3.1. What has been already done in the scope?

Currently implemented algorithms for muon momentum regression in the Level-1 trigger of the CMS includes a mixture of neural nets and decision trees. The algorithms used are elaborated further below.

a. Boosted decision trees

Decision trees are weak learners of complex data. Hence, gradient boosting algorithms are applied to make ensembles of models. Boosted decision trees are efficient on both regression tasks and are more accurate compared with random forests. However, they have tendencies to overfit and are too sensitive to outliers.

b. Fully connected deep neural networks

Fully connected or dense neural networks are connected to every single input parameter. They tend to have no pooling layers or dropout layers. They are also very demanding on the hardware resources and time constraints when compared with convolutional neural networks. They get reasonably good accuracy and can approximate almost any function. They ingress data as a tensor.

c. 1D Convolutional neural networks

1D convolutional networks are similar to dense neural networks but they have one important characteristic. They have a kernel filter that can make convolutional feature maps much faster. They look at a subset of data in an axis, unlike dense neural networks. AlexNet [6] from the University of Toronto and VGG [7] from the University of Oxford are prime examples of optimized use of convolutional neural networks. Convolutional neural networks consist of feature maps, pooling layers, dropout layers and they also give insights from their saliency maps. Visual understanding of CNNs has been thoroughly explored [8].

3.2. Project ideas mentioned in the project description

a. Benchmark graph neural networks for momentum regression in the trigger system

Graph neural networks are a modern twist on convolutional neural networks [9]. They are generally used for irregular data. The idea with GNNs are that we can make the network learn the structure of data utilizing their node and edge composition. Nodes are used to predict information about unlabeled nodes based on labeled nodes. Edges are used to predict information about new edges based on labeled edges. One of the most useful graphs to ingress models would be a Bayesian graph as they are good for inferencing.

b. Develop deep learning compression algorithms for optimized inference

One of the issues current deep neural models are facing is the model size. Models are becoming increasingly complex and large. This makes training time, and inference time to increase. Hence, we need special algorithms when running deep neural networks in resource-constrained environments. This leads to compression algorithms for optimized inferences. A few compression algorithms are listed below.

1. Pruning Neural networks – means to delete neurons in a layer that don't meaningfully contribute to the training or inference process. Modern works facilitate the pruning process dynamically while runtime [10].

2. Deep compression [11] – introduced in 2016 shows a novel way to decrease the network size without affecting the accuracy and the precision of the network. The paper introduces a 3-phase process. The method first prunes the network by learning only the important connections. Next, the method quantizes the weights to enforce weight sharing. Finally, the method uses Huffman coding. After the first two steps, the authors retrain the network to fine-tune the remaining connections and the quantized centroids. Pruning reduces the number of connections by 9 to 13 times. Quantization then reduces the number of bits that represent each connection from 32 to 5.

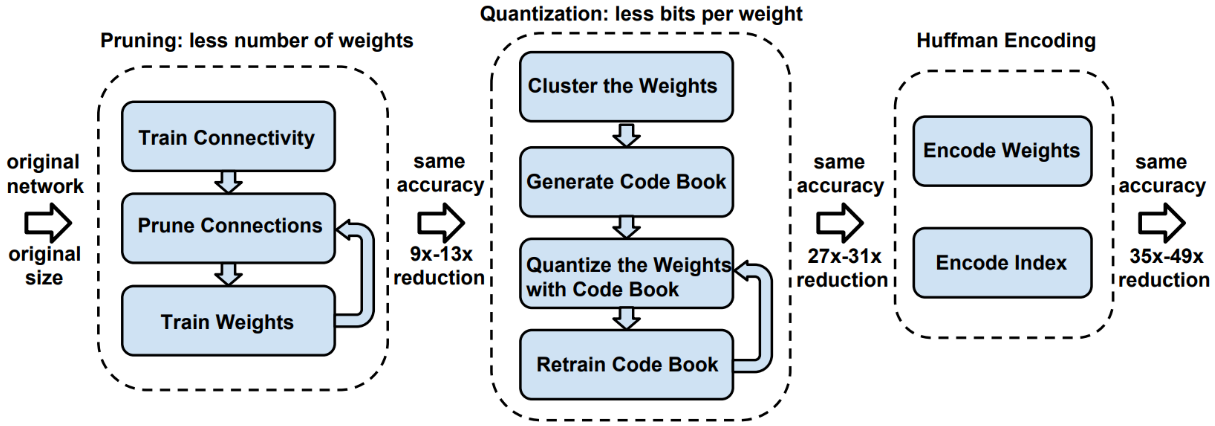

Figure 3 - Deep compression process

3. Data Quantization [12] – The authors proposed a dynamic precision data quantization method to improve bandwidth and resource utilization. The method proposed demonstrates remarkably shorter representations of models with still achieving comparable accuracy. Their work is best demonstrated on Field Programmable Gate Array or FPGA.

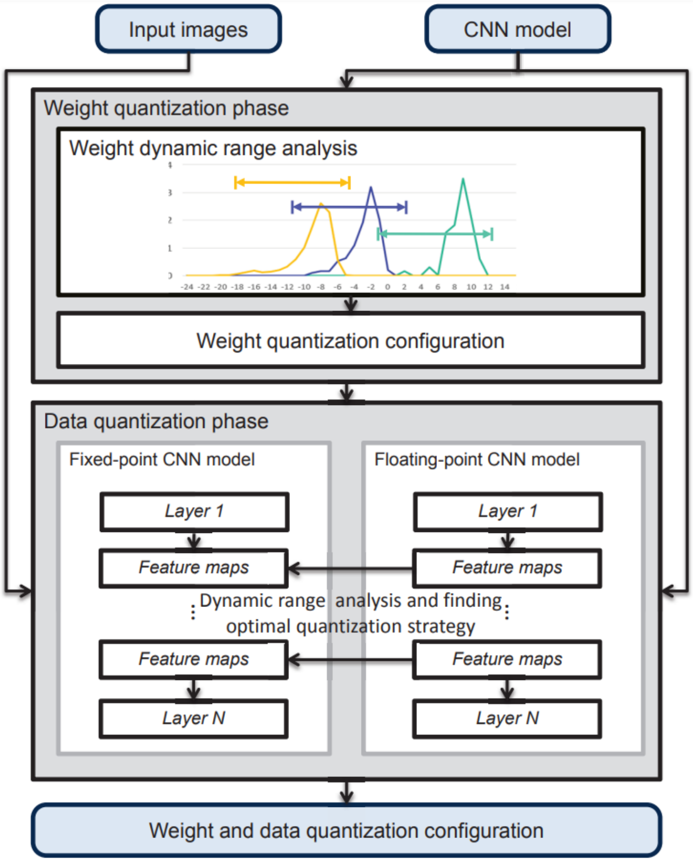

Figure 4 - Data Quantization in FPGA

4. SqueezeNet [13] - Smaller models have faster inference. This work demonstrates AlexNet level accuracy on imagenet dataset with 50 time less parameters. This model compresses SqueezeNet to less than 0.5 MB in size (more than 500 time the original size).

5. SqueezeDet [14] – SqueezeDet is a derivative of SqueezeNet. This work hones its skills towards the real-time inference speed of high accuracy models. This is used in autonomous cars. This is not only faster but also smaller and consumes less energy than previous models. The above models SqueezeNet and SqueezeDet are ideal for deploying in FPGAs and other resource constrained hardware.

c. Prototype emulation of deep learning-based trigger inference within practical latency requirements

Prototype Emulator of EMTF in the Level-1 trigger system has to be developed. It has to conform to the practical latency constraints. The time constraints for the model inference is about 500 nanoseconds. Memory constraints are generally negligible.

$$time\ constraint = 500\ x\ 10^{-9}\ seconds$$

3.3. Project ideas found through cursory research

- Since we need to infer at the level-1 trigger in real-time, we could investigate the data collection methodology and do an exploratory data analysis (EDA) on the real-time data using Apache Flink. Apache Flink is an opensource real-time data stream processing engine.
- Since the data from CMS is converged from the Level-1 trigger, and high-level trigger, to get the final dataset. We have 2 representations of the same data. Autoencoders are great at converging data and de-converging (diverging) them back again. If we have enough training data to train an autoencoder, we could effectively use an Autoencoder to carry out the jobs of both the Level-1 trigger and the High-Level trigger.
- Extreme Gradient Boosting Regressor algorithms can be explored as well. (XGBoost)

4. Project Evaluation

4.1. Neural Networks evaluation criterion

Regression has its own criteria for evaluation. Regression tasks in machine learning make use of the following metrics of validation as evaluation of performances [15]. These are the evaluation metrics mentioned in the sci kit learn documentation for regression tasks.

a. Explained variance score
b. Max error
c. Mean absolute error
d. Mean squared error
e. Mean squared log error
f. r^2 score
g. Mean poisson deviance
h. Mean gamma deviance
i. Mean tweedie deviance

4.2. Project Evaluation criterion

The project has its own evaluation criteria as mentioned in the project task description.

a. Functional prototype emulator for level-1 trigger using FCNN
b. Functional prototype emulator for level-1 trigger using CNN and GNN
c. Benchmarks of deep network model inference for muon momentum regression for prompt and displaced muons.

5. Deliverable Artifacts

The following are deliverable artifacts from this project

a. Prototype emulator for level-1 trigger task of muon momentum
b. Report on functional benchmarks for inference speed and accuracy of deep neural models.
c. Documentation for the developed artifacts.
d. Starting guide to facilitate the use of emulator.

6. Mentor Communication

- Mar 27th 2020. I sent the mentors first draft of the proposal to get their feedback and subsequently requested feedback from Sergei Gleyzer and Ali Hariri. They also gave me some tasks to be done.
- On March 28th I clarified tasks submission details from them again and requested a clarification in the task.
- On March 29th Ali Hariri replied with a well-advised feedback on the datasets, Sergei Gleyzer clarified my doubt on a physics theory, and gave a general overview of the project in the draft proposal.
- On March 30th I completed the 3 tasks assigned to me and submitted them to my github repository link.
- On March 30th I received feedback on my draft from the mentor Ali and I was able to finalise the submission copy.

7. Results to the Particle Physics community

This project is to principally make an emulator for the level-1 trigger for the CMS experiment. This prototype will help the physicists and engineers to carry out tests without actually expediting the cumbersome process of deploying the particle accelerator for a simple task of ratifying the software. This project will also be giving a report on the performance benchmarks of various models such as FCNN, CNN, and GNN in addition to decision trees.

8. Other Commitments

In the 2nd week of August, I will be traveling to a conference at the University of Oxford to present my paper. I would be unavailable for about a week. To compensate for that, I will begin my testing and ratification a week earlier, parallelly getting feedback from the mentors. I will start working on the project a week before the community bonding period ends,

Besides, I will have a cushioning week as specified in the timelines below to do any remaining tasks. Other than the above I do not have any upcoming deadlines, and I can fully commit myself to the project.

9. Timelines

9.1. Summarized Timeline

I have compartmentalized the task into sub-tasks, based on the evaluations for easy comprehension. Before the first evaluation officially we have 5 weeks, however since I will be starting a week earlier, I hope to finish the set of objectives as shown below.

a. First Evaluation – Benchmark various assorted algorithms for muon momentum estimation regression, research and develop a suitable compression algorithm for the deep neural model. **(5 + 1 weeks – starting 1 week earlier)**
b. Second Evaluation – Contd. research and develop a suitable compression algorithm for the deep neural model, Prototype emulator for the inference. **(3 weeks)**
c. Final Evaluation – Contd. Prototype emulator for the inference, Starting guides for the emulator for inference. **(4 weeks)**

9.2. Detailed Timeline

Based on the above is the detailed timeline

Note-1: I will have one week for introspection before the first evaluation to think about the work done in the previous weeks to see if it can be improved.

Note-2: I will be using Trello to keep track of my tasks unless specified otherwise.

a. First Evaluation
   1. Week 1 – investigate the existing codebase, explore how the codebase works skeumorphically with the CMS trigger.
   2. Week 2 – research benchmarking metrics for various deep learning algorithms as a function of time
   3. Week 3 – benchmark various deep learning algorithms as a function of time
   4. Week 4 – introspection week
   5. Week 5,6 – research and implementations of deep neural model compression algorithms

b. Second Evaluation
   1. Week 1 – contd. implementations of deep neural model compression algorithms
   2. Week 2 – implement emulator prototype
   3. Week 3 – contd. implement emulator prototype, begin writing starting guides

c. Final Evaluation
   1. Week 1 – complete documentation and diagnostic reviews
   2. Week 2 – complete starting guides
   3. Week 3 – diagnose and complete the models and emulators for any discrepancies
   4. Week 4 – additional week for cushioning any unforeseen issues

9.3. Gantt Chart

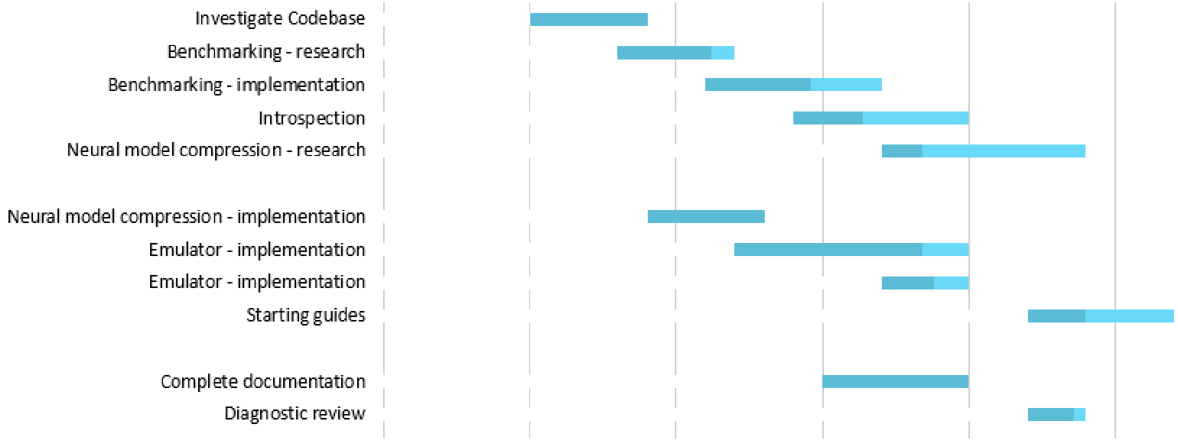

Figure 5 - Gannt Chart for the timeline

10. References

[1] D. Acosta *et al.*, “Boosted Decision Trees in the Level-1 Muon Endcap Trigger at CMS,” *J. Phys.: Conf. Ser.*, vol. 1085, p. 042042, Sep. 2018, doi: 10.1088/1742-6596/1085/4/042042.

[2] L. Taylor, “CMS: Higgs boson decays to four muons,” *Physics World : Dec 1998*, 01-Oct-1997. https://cds.cern.ch/record/39444 (accessed Mar. 26, 2020).

[3] L. Cadamuro, “The CMS Level-1 trigger system for LHC Run II,” *J. Inst.*, vol. 12, no. 03, pp. C03021–C03021, Mar. 2017, doi: 10.1088/1748-0221/12/03/C03021.

[4] A. Bocci, “The Trigger of the CMS Experiment,” p. 56.

[5] D. Acosta and U. Florida, “HL LHC Muon Trigger Upgrade Overview,” p. 40.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: 27-Mar-2020. [Online]. Available: http://arxiv.org/abs/1409.1556.

[8] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” *arXiv:1311.2901 [cs]*, Nov. 2013, Accessed: 19-Sep-2019. [Online]. Available: http://arxiv.org/abs/1311.2901.

[9] J. Zhou *et al.*, “Graph Neural Networks: A Review of Methods and Applications,” *arXiv:1812.08434 [cs, stat]*, Jul. 2019, Accessed: 27-Mar-2020. [Online]. Available: http://arxiv.org/abs/1812.08434.

[10] J. Lin, Y. Rao, J. Lu, and J. Zhou, “Runtime Neural Pruning,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2181–2191.

[11] S. Han, H. Mao, and W. J. Dally, “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” *arXiv:1510.00149 [cs]*, Feb. 2016, Accessed: 27-Mar-2020. [Online]. Available: http://arxiv.org/abs/1510.00149.

[12] J. Qiu *et al.*, “Going Deeper with Embedded FPGA Platform for Convolutional Neural Network,” 2016, pp. 26–35, doi: 10.1145/2847263.2847265.

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” *arXiv:1602.07360 [cs]*, Nov. 2016, Accessed: 27-Mar-2020. [Online]. Available: http://arxiv.org/abs/1602.07360.

[14] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, “SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving,” *arXiv:1612.01051 [cs]*, Jun. 2019, Accessed: 27-Mar-2020. [Online]. Available: http://arxiv.org/abs/1612.01051.

[15] “3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 0.22.2 documentation.” https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics (accessed Mar. 27, 2020).