**Comparative study of Machine learning frameworks**

|  |  |  |  |
| --- | --- | --- | --- |
| Jeff Blackburn School of Computer Science and Statistics Trinity College Dublin Ireland blackbuj@tcd.ie | | Abhishek Jain School of Computer Science and Statistics Trinity College Dublin Ireland jainab@tcd.ie | Siddharth Dillon School of Computer Science and Statistics Trinity College Dublin Ireland dhillons@tcd.ie |
|  |

***Abstract* – last**

**aaaaaaa**

***Index Terms* -**

**aaaaaaa**

# Introduction

Introduce a) the background of your work, and clearly state b) the research problem and c) your research question/goal (do not just copy & paste the question from our project list. Think carefully about the problem and question and phrase it accordingly).

# Related Work

Explain and discuss critically what other researchers did to answer the research question and what their results were. Do not spend too much time in finding related work but focus on describing properly the work you found. We will not mark you down when some important related work is missing (as long as you describe some related work).

# Methodology

Briefly describe the dataset used, including the format of the data, and how the data was processed. Explain how and why you (pre-)processed the data to make it suitable for your analysis. Describe the machine learning algorithms selected and how you went about selecting appropriate values for the algorithm parameters. Present plots justifying your choices and discuss your decisions. Given the limited time you have, we do not expect a perfectly tuned system. Rather it is the critical discussion here that is important, and this should cover the major issues affecting your choices plus the level of uncertainty that your analysis indicates for the parameter choices. Also, explain and justify how you evaluate your work (e.g. chosen metrics, how training and test data was split …).

# Results & Discussion

For comparing results accross different framworks and different size of dataset we have used evaluation metrics like MAE, RMSE, MSE for Linear rregression. Accuracy, confusion metrics and AUC for Logistic Regression and KNN.

Implementation of linear regression produced similar results in different metrics, but there was a significant difference in training time as Sci-Kit Learn was able to train linear regression model in 0.07 seconds, tensorflow took 3 seconds whereas PyTourch completed training in 11 seconds for our medium size dataset. Tensorflow and Sci-Kit learn training time was almost same for different size of dataset whereas PyTourch training time increased when size of data increased.

For Logistic Regression model we used evaluation metrics like accuracy, Error rate, FNR, FPR, AUC and Recall/TPR. Logistic regression implementation produced similar results in all frameworks. For this algorithm scikit learn was faster when compared to Tensorflow and PyTourch. Both Tensorflow and Sci-Kit Learn maintained almost constant training time accross different size of datasets

For KNN comparison we used evaluation metrics like Error rate, AUC and confusion matrix. PyTourch offered slightly better accuracy of 0.6 compared to scikit learn which offered 0.5 for our small dataset, but when compared with medium and larger dataset it was similar for both framework. AUC and confusion metric produced similar results for scikit pyTourch and tensorflow, Training time was similar when using Sci-Kit Learn across dataset of different size.Tensorflow took 2.12 and 3.96 seconds for small and medium dataset respectively but when used for training large dataset it took 17 Seconds whereas PyTourch took 62 Seconds.

# Limitations & Outlook

Sci-Kit Learn was able to perform better in term of training time and accuracy than tensorflow and Pytourch in our implementation of Linear regression, Logistic Regression and KNN. Altought further comparison can be carried out in implementation of neural networks using tensorflow and PyTourch. In terms of GPU support Tensorflow and PyTourch provide out of the box support for GPU while Sci-kit Learn do not provide support for GPU [1]. Also Tensorflow provide specialized library for web development - Tensorflow.JS and Tensorflow Lite for mobile and embedded devices. both tensorflow and pytourch provide advance features like Distributed training which can provide significant training boost when dealing with giant datasets and models.

# References

[1] Http://scikit-learn.org/stable/faq.html#will-you-add-gpu-support

**A HEADINGS IN APPENDICES**

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the appendix environment, the command section is used to indicate the start of each Appendix, with alphabetic order designation (i.e., the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure within an Appendix, start with subsection as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

**A.1 Introduction**

**A.2 Experimental and Computational Details**

*A.2.1 Sample Fabrication*

*A.2.2 Quasi-Static Measurements: MOKE and MFM*

Component Structures

Magnetization.

*A.2.3 Dynamic Measurements: BLS*

*A.2.4 Ground-State Magnetization Determination and DMM Micromagnetic Simulations*

Determined.

Micromagnetic

**A.3 Results and Discussion**

*A.3.1 Magnetization Curves and MFM Characterization*

*A.3.2 Field Dependent BLS Measurements and DMM Calculations*

*A.3.3 Analysis of the Dynamic Coupling as a Function of the Gap Size*

**A.4 Conclusions**

**A.5 References**