TensorFlow vs PyTorch: Comparative Analysis of Algorithm Performance in Modern Machine Learning

Project by Sharelle David
Supervisors: Dr. Mahsa Zolfaghari, Dr. Maryam Shahpasand

1. Introduction

This project analyzed the performance of two machine learning frameworks using python. I produced an image classification model by developing a hybrid Convolutional Neural Network, deployed the algorithm onto TensorFlow and PyTorch. Testing the algorithm performances. TensorFlow and PyTorch models, particularly with advanced architectures like CNNs, was a complex experience to achieve a grasp on Application Programming Interface API. Hyperparameter tuning and optimization was implemented, the results I achieved a latency reduction of 8.2% per image.

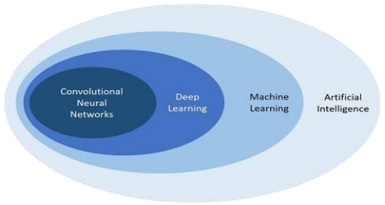


Figure 1 – Diagram of AI Subsets

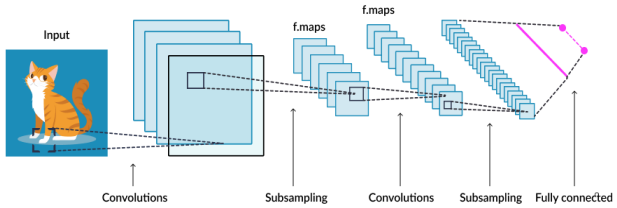


Figure 2 – Example of CNN Layers

2. Research

TensorFlow - open-source library for deep neural networks, flexible architectures for training machine learning models and symbolic computational graphs that represent mathematical operations and data flows and nodes. Simple built-in high-level API. Visualizing training with Tensor board library. [1]

PyTorch - an open-source machine learning library using python programming typically used for developing neural network models. Python-like coding. Dynamic graph. Easy and quick editing. Good documentation and community support. [2]

CNN - A convolution neural network is specifically designed for processing grid data images. Techniques used: Data Augmentation, Batch Normalization, Dropout, Interpreting/Visualizing, Object Localization & Detection, Adversarial Training.[3]

3. Design

TensorFlow and PyTorch were used as Deep Learning frameworks. ResNet50 CNN model and CIFAR10 dataset used as the architecture and dataset for the image classification CNN. The environment for developing the CNN for each framework was Google Colab.

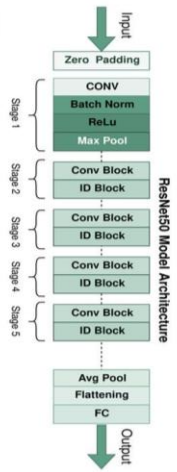
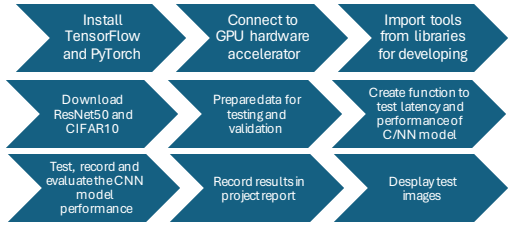


Figure 3 – ResNet50 CNN Model Architecture

4. Testing



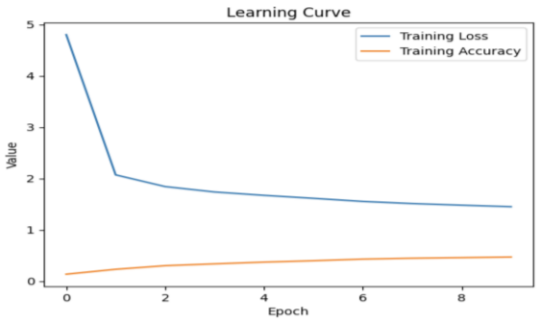


Figure 5 – TensorFlow CNN Performance

5. Results

Deep Learning Framework	Average Iteration	Loss of Function	Test Accuracy	Latency per Image
TensorFlow	44.4 Seconds	1.3959	49.96%	0.02 Seconds
PyTorch	33.7 Seconds	1.3978	48.67%	0.01 Seconds

6. Solution

To improve my evaluation of the two libraries performances, CIFAR100 would have been used instead of CIFAR10 as the dataset. The benefits of training the hybrid convolution neural network with a larger dataset would have given the deep learning frameworks more processing to do and the latency and well as accuracy metrics more of a challenge to truly test the deep learning frameworks capabilities and processing time. Hyperparameters such as quantization to be introduced to improve testing process and overall performance metrics.

7. References

1. F. Chollet, 2022, 'Deep Learning with TensorFlow and Keras', Available at: https://books.google.co.uk/books?hl=en&lr=&id=iq6REAAAQBAJ&oi=fnd&pg=PP1&dq=tensorflow&ots=ujuuuU4pQI&sig=BVUJ9Y_mbqpqRt0XrdbZyBfEHuY&redir_esc=y#v=onepage&q=tensorflow&f=false Accessed on 13/04/2024.

2. S. Chintala, 2020, 'Deep Learning wit PyTorch', Available at: https://books.google.co.uk/books?id=BjszEAAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false Accessed on 15/04/2023

3. Z. Zhao, 2023, 'ressearch on Text Generation Model of Natural Language Processing Based ov Computer Artifical Intelligence', Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10257788 Accessed on 13/04/2024