

**VIETNAM NATIONAL UNIVERSITY
HO CHI MINH UNIVERSITY OF SCIENCE**



**COURSE PROJECT REPORT:
NEURAL NETWORKS FOR CLASSIFICATION**

**22127045 VU MINH CHIEN
22127120 CAO NGUYEN HUY HOANG**

INSTRUCTORS

**Dr. Bui Duy Dang
M.S. Nguyen Thanh Tinh
M.S. Tran Trung Kien**

SUBJECT: INTRODUCTION TO MACHINE LEARNING

HO CHI MINH CITY – 2024

Table of contents

1. Self-evaluation of assignment requirements:	2
2. Analysis and comparison of the libraries and frameworks:	3
2.1. Training Time:	3
2.2. GPU Memory Usage:	4
2.3. Pros and cons of each library/framework:	5
3. Reference:	7

1. Self-evaluation of assignment requirements:

We have built and trained models from the following libraries and frameworks:

- Scikit-learn (MLPClassifier).
- Tensorflow/Keras.
- PyTorch.

Moreover, each individual has taken part in and accomplished the following specifications:

Tasks	Name	Percentage
Data Preparation	Vu Minh Chien	100%
Model Design	Cao Nguyen Huy Hoang	100%
Selecting Loss Function and Optimizer	Cao Nguyen Huy Hoang	100%
Model Training	Vu Minh Chien	100%
Model Evaluation	Vu Minh Chien	100%

2. Analysis and comparison of the libraries and frameworks:

2.1. Training Time:

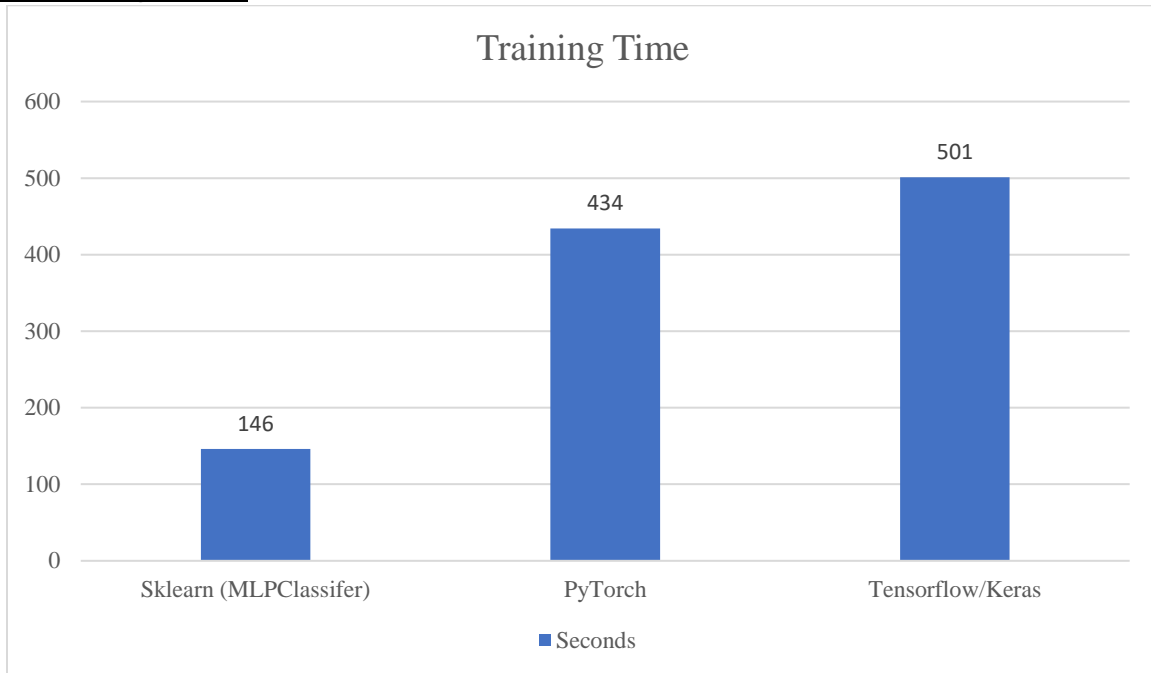


Figure 1: Training time of 3 libraries and frameworks

- Analysis:
 - Scikit-learn has the fastest runtime with 146 seconds, this is only because it only does around 12 or 13 iterations before stopping due to it not improving more than 0.001 for more than 3 consecutive epochs.
 - PyTorch has the second-best runtime with 434 seconds. It generally works faster and provides superior debugging capabilities compared to Tensorflow/Keras. [1]
 - Tensorflow/Keras is the slowest out of the 3 models with 501 seconds.
- Comment:
 - Pytorch has higher performance than Keras in terms of training time.
 - Sklearn performs the best out of the 3 libraries/frameworks with the validation scores unchanged at 0.01 tolerance after 3 consecutive epochs.

2.2. GPU Memory Usage:

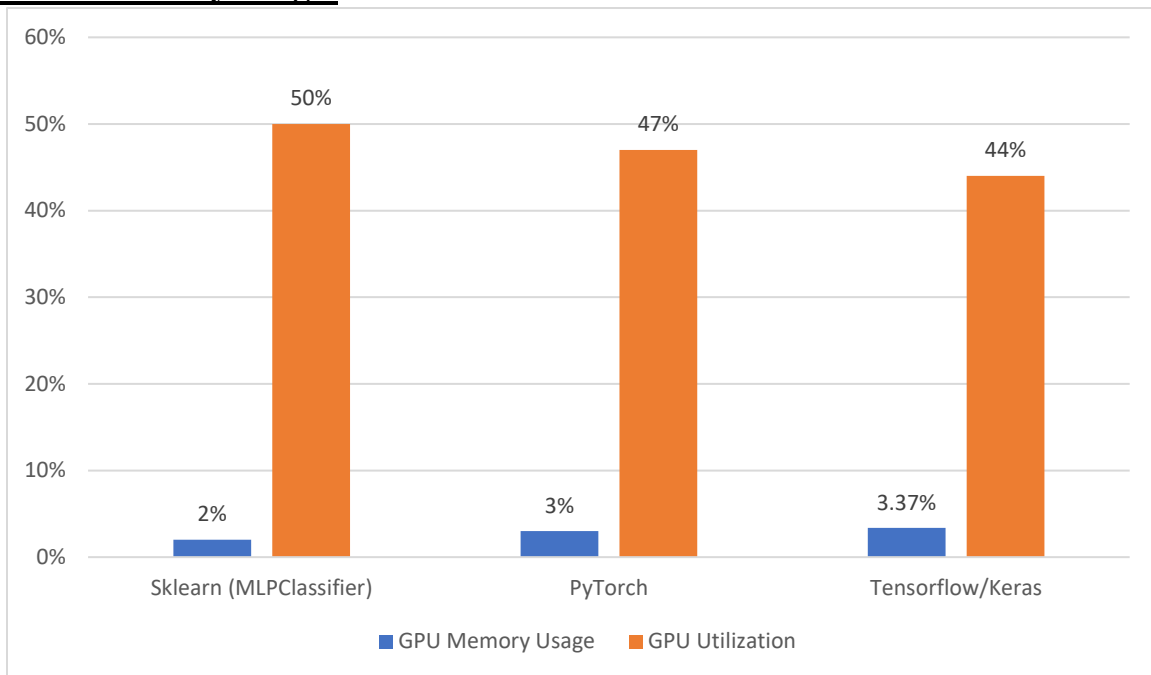


Figure 2: GPU Memory Usage and Utilization of 3 libraries and frameworks

- Analysis:
 - Scikit-learn has the lowest GPU memory usage (2%). This can be because Scikit-learn isn't specifically built to utilize GPU but rather CPU.
 - PyTorch and Tensorflow/Keras have almost similar GPU memory usage (3 and 3.37% respectively) due to them being exclusively designed to exploit GPU acceleration to perform faster.
 - Each library/framework has a 3 percent difference in GPU memory utilization, which are 50%, 47% and 44% for Scikit-learn, PyTorch and Tensorflow/Keras in the order given. 50% memory utilization from Scikit-learn can be a result of the GPUtil library reserving that memory. The percentage shown by PyTorch and Tensorflow/Keras may stem from their memory allocation, both dynamically allocated.
- Comment:
 - Tensorflow/Keras has a slightly lower GPU utilization than but a higher max GPU memory usage than PyTorch.
 - Scikit-learn is used on CPU rather than GPU.

2.3. Pros and cons of each library/framework:

- Scikit-learn:
 - Pros:
 - Is a machine learning library that mostly utilizes Numpy.
 - Does not use GPU acceleration, which means that it can be accessible to a wider range of users who do not have GPU available.
 - Is user-friendly.
 - Has a lot of machine learning algorithms. [2]
 - Cons:
 - No GPU acceleration means that it is going to take longer if the data input size is large.
 - Can easily be overfitting.

- PyTorch:
 - Pros:
 - Fast runtime, lower GPU memory usage than Tensorflow/Keras.
 - Offers GPU acceleration for better performance.
 - Can be used for large data sets. [1]
 - Highly customizable.
 - Cons:
 - Lesser well-known than Tensorflow/Keras.
 - Doesn't have some built-in class or function like early stopping available.
 - Its architecture is rather complex and not easy to read.

- Tensorflow/Keras:
 - Pros:
 - User-friendly interface.
 - Have built-in features like early stopping.
 - Can be used with fit() function to train the model.
 - Easily readable architecture. [1]
 - More popular than Pytorch.
 - Cons:
 - Slower runtime and higher GPU memory usage than PyTorch.
 - Best used in smaller datasets.

2.3. Ease of Use:

Scikit-learn (MLP Classifier)	PyTorch	Tensorflow/Keras
User-friendly interface, have built-in machine learning algorithms.	Have to define more classes and manually create the loop.	User-friendly interface, easy to use.

3. Reference:

[1]. <https://www.datacamp.com/tutorial/pytorch-vs-tensorflow-vs-keras>, 25/12/2024

[2]. <https://stackoverflow.com/questions/54527439/differences-in-scikit-learn-keras-or-pytorch>, 24/12/2024