An Evaluation of Machine Learning Frameworks

Franck Wafo
Hochschule Ruhr West
Bottrop, Germany
Franck.WafoNguembu@stud.hs-ruhrwest.de

Ivan Cedric Mabou
Hochschule Ruhr West
Bottrop, Germany
ivan.mabououabo@stud.hs-ruhrwest.de

Dan Heilmann
Hochschule Ruhr West
Bottrop, Germany
dan.heilmann@stud.hs-ruhrwest.de

Nico Zengeler
Hochschule Ruhr West
Bottrop, Germany
nico.zengeler@hs-ruhrwest.de

Uwe Handmann
Hochschule Ruhr West
Bottrop, Germany
uwe.handmann@hs-ruhrwest.de

***Abstract*—Artificial Intelligence (AI) and Machine Learning (ML) have become increasingly important for any organization that wants to stay competitive and speed up its processes. However, while organizations can choose from a variety of machine and deep learning (DL) frameworks, it is important to remember that these frameworks serve very different purposes. Therefore, the choice of a framework adapted to your needs is a decision of the utmost importance. In this article, we present an evaluation of some of the most popular machine and deep learning frameworks developed, based on an image recognition task.**

I. Introduction

Our research focuses mainly on the evaluation of the following frameworks: Tensorflow, Theano, and PyTorch. This should provide machine learning practitioners and researchers with answers as to which framework best suits their needs. Moreover, since the scope of these frameworks is very broad, we decided to limit their application to image classification. For the purposes of this evaluation, we identified the following criteria, such as model loss and accuracy, training time, confusion matrix or accuracy rate obtained for each of the ten classes from our data set, the degree of popularity observed by the survey conducted by KDnuggets, as well as by the results obtained via Google Trend. Following these criteria, our results suggest on the one hand that, as far as performance is concerned, the three frameworks are relatively similar and reach quite high levels of accuracy, all exceeding 90%. On the other hand, in terms of the general interest surrounding these frameworks, our results showed that Tensorflow remains the preferred platform for users. However, we also observed that this interest is gradually shifting towards PyTorch.

Essentially, we have divided our paper into five sections. The second section reports on previous studies comparing different machine learning frameworks. The third section describes the research method used in our study to evaluate the three selected frameworks. Then, in the fourth section, we present the results of this evaluation. Finally, in the fifth section, we conclude our work and discuss our findings.

II. Related work

Gevorkyan et al. compare the following five libraries: Keras, Scikit, TensorFlow, PyTorch and Theano, focusing on the example of a multi-layered perceptron, which is applied to the problem of handwritten number recognition. The duration of the study is compared as a function of the number of epochs and the accuracy of the classifier [3]. Their results show that almost all libraries, with the exception of PyTorch, have approximately the same learning time. They explain that in the case of PyTorch, the longer learning time can be explained by the support of a dynamic calculation graph, which apparently imposes additional calculation costs. In addition, they found that the TensorFlow library had an average accuracy result, behind PyTorch and Theano [3].
Dinghofer and Hartung present a comparative overview of TensorFlow, Keras, PyTorch and Caffe with a focus on computer vision tasks and applications based on the following criteria: Features and uses, as well as adoption and popularity [2]. They found that of the various DL frameworks suitable for computer vision applications, none clearly stand out from the others. Further, they report that although TensorFlow is currently the dominant framework, other frameworks are more advantageous in terms of ease of entry, collaboration features and speed [2].
Stanin and Jovi compare nearly 20 free python-based libraries and identify the various advantages and disadvantages of python-based libraries, and separates them into six main groups: core libraries, data preparation, data visualization, machine learning, deep learning and big data [7]. Libraries that we are interested in with regards to our study are clustered under deep learning. These have been analyzed according to the criteria layers, losses, activation function, optimizers and GPU acceleration. Their results show on the one hand that the Caffe is not only not well documented, but also has only basic functionality. On the other hand, Pytorch, Keras and TensorFlow have a well-structured documentation. In addition, they note that TensorFlow has many more features than the other libraries [7]. Further, the authors suggest to adopt TensorFlow for

agile projects with a high need for customization, and PyTorch or Keras for rapid prototyping [7].
Simmons and Holliday compare two of the Frameworks, Pytorch and TensorFlow using a binary classification problem and evaluate them according to the training time, memory usage, and ease of use [6]. The results show that the two Frameworks have more or less similar performances in terms of precision. Nevertheless while TensorFlow is more convincing in terms of training time, PyTorch shows more interesting results with regards to memory usage. And as for the previous study PyTorch would be more adequate for quick prototyping, and TensorFlow for projects which with a high need for customization [6].

III. Research Method

The primary goal of this study is to provide an evaluation of three of the most popular frameworks for machine learning: TensorFlow, Theano and PyTorch. However, although we could have chosen to evaluate frameworks other than these three, there were two main reasons for our choice. Firstly, TensorFlow and PyTorch are very popular in the machine learning community. Secondly, Tensorflow and Theano are possible backends for Keras, so we were able to reuse the same code for both frameworks. In addition, it should be noted that Keras has three backend implementations, including Tensorflow, Theano, and CNTK. However, unlike the other two, CNTK was not considered for this study. Indeed, Kaggle, which we used as a working environment, did not have access to this library.
Deep Learning is a subset of machine learning algorithms inspired by the structure and function of the brain called artificial neural networks. This is a back-propagation based learning algorithm, which corrects errors automatically. Deep learning again knows a variety of different models like:

- Feed Forward Networks (FFN)
- Convolutional Neural Networks(CNN)
- Recurrent Neural networks(RNN)
- Generative Adversarial Neural Networks(GAN)
- (Variational) Auto-Encoders (VAE)

Our evaluation is based on the subject of image classification. Basically, these images are data in the form of 2-dimensional matrices. Image classification is the method of assigning an input image of one kind from a fixed set of categories and classify those using different algorithms. It is similar to data classification in machine learning. In both image and data classification we are playing with number/digits. The human eye perceives an image as a set of signals which are processed by the visual cortex in the brain but computer cant. Computer perceives a vector image or sequence of pixels with discrete numerical values.

A. Dataset Description

For the purpose of our study, we used the Fashion-MNIST dataset available on Kaggle. The latter is the worlds largest data science community with powerful tools and resources to help you achieve your data science goals. The Fashion-MNIST dataset contains different clothing images of 60,000 training set and 10,000 testing sets of ten classes. It is standard dataset used in computer vision and deep learning. The mapping of all 0-9 integers to class labels is listed below:

- 0: T-shirt/top
- 1: Trouser
- 2: Pullover
- 3: Dress
- 4: Coat
- 5: Sandal
- 6: Shirt
- 7: Sneaker
- 8: Bag
- 9: Ankle boot

Our study addresses a multi-class classification problem, i.e. there are more than two classes to be predicted. Here we need to classify ten different clothing items (target labels are of integers, ranging from 0 to 9). In our dataset pixel values for each image in the dataset are unsigned integers in the range between 0 and 255. Basically, the inputs used in our study are represented by 28x28 images and the outputs by ten product categories.

B. CNN Model

In this study, we have chosen a convolutional neural network (CNN) model. It is one of the most popular models used in deep learning [1]. The CNN model was originally designed for image processing. It reduces the number of parameters to be learned and the amount of computation performed in the network increasing the efficiency of the model. Thus, it is more suitable than the other models cited above for the realization of our study. Note also that this model has proven to be very efficient in other areas of machine learning such as natural language processing (NLP) [8]. The figure below illustrates the different layers that compose our model.
In addition, to compile our model, we defined Categorical CrossEntropy as the loss function and Adam as the optimizer. Categorical CrossEntropy calculates the loss between labels and predictions. This function is used when there are two or more label classes to be predicted. Therefore, it was the most appropriate for our study. However, while this function is available for Tensorflow and Theano, we were unable to find its equivalent for PyTorch. As a result, we had to implement it ourselves.

C. Evaluation Criteria

The criteria used to assess the three frameworks were organized into two categories: Internal and External Criteria. The first group refers to criteria related to the evaluation of the model. Examples in this category include model loss and accuracy, training time, and confusion matrix. The second group encompasses criteria used to gauge the usage, interest and popularity of the framework such as the KDNuggets Usage Survey, and Google search activity.

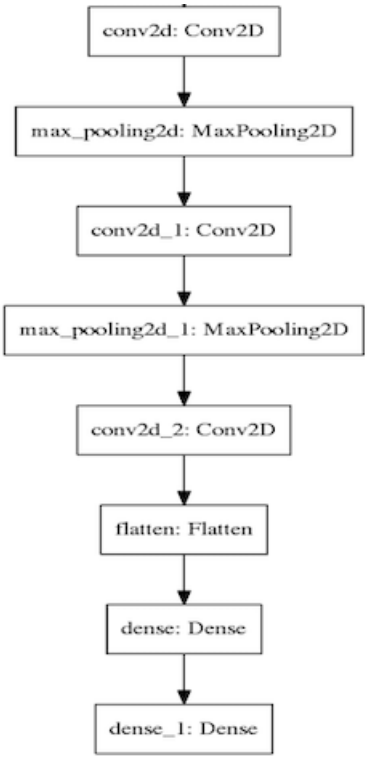

Fig. 1. CNN Model

IV. Results

A. Internal environmental criteria

1) Model Loss and Accuracy: We used the accuracy and loss of the model, as well as the training time and the confusion matrix to assess the performance of the above-mentioned frameworks. As described in the previous section, we have divided the training data set into two data sets: the train and the validation set. Therefore, we differentiate between the accuracy and loss of the training and validation set, and the accuracy and loss of the test set.
2) Training and Validation Accuracy and Loss: For each framework, as shown in the figure above, we have displayed the graphs that imply the accuracy and loss on the y scale, and the number of epochs on the x scale. The accuracy and loss of the model on the training set is indicated by a blue curve, while the accuracy and loss on the validation set is indicated by an orange curve.
Model accuracy on the training set is higher with Tensorflow and Theano at 99.34% and 99.18% respectively. PyTorch achieved an accuracy of 94.31%, which is also quite high, but less than the other two. Furthermore, while on the same set, Tensorflow and Theano had loss values very close to each other with 0.019 and 0.025 respectively, PyTorch achieved a significantly higher value with 1.52.
Then, over the validation set, the model accuracy was quite similar with the three frameworks reaching 90.65% and 90.90% for Tensorflow and Theano respectively, and 91.01% for PyTorch. However, regarding the model loss on this set, we have observed with Tensorflow and Theano that from the 5th epoch onwards, it stops decreasing, and starts increasing again. Indeed, the validation loss on the two frameworks increases to 0.69 and 0.65 respectively. This is an over-fitting problem which is mainly due to the fact that we did not regularize the data and/or use dropout layers in order to have a similar loss function for all three frameworks. Therefore, although removing the dropout layers helped us to implement a loss function for PyTorch that would be equivalent to the categorical CrossEntropy of Tensorflow and Theano, it also created the overfitting problem for the latter. PyTorch, on the other hand, achieved a loss in value of 1.55 without being overfitted.
The accuracy of the model was also very similar in the test set for all three frames, which all achieved an accuracy of around 91%. Then, with regard to the loss of the model on this set, the values obtained with the three frameworks were very similar to the one observed on the validation set. Indeed, Tensorflow and Theano reached 0.66 and 0.59 respectively, which is slightly lower than the values obtained with the same frames on the validation set. The loss value achieved with PyTorch remained the same as that on the validation set at 1.55.
With respect to training time, which is another criterion of our evaluation, we find that with Theano, the model took less time to learn than with the other two frameworks where it required about the same amount of time (see table below).
3) Confusion matrix: We can see that for the three frameworks, the model reaches quite similar levels of precision for each class. It should also be noted that the classes number 0, 2, 3, 4 and 6 representing respectively the objects: T-shirt/top, pullover, dress, coat and shirt obtained a precision rate lower than 90% for the three frameworks, in particular the class number 6 corresponding to the shirt. This can be explained by the fact that these objects have many similarities and the algorithm confuses them more frequently during the training. This is the reason why the accuracy rate of these classes is lower than the others. Nevertheless, it would be difficult to compare the frameworks from the point of view of the predicted classes, because although the accuracy rates of the classes are slightly different, they remain relatively similar.

B. External environmental criteria

External criteria illustrate to what extent a ML/DL framework is popular among researchers and developers. The KDNuggets Usage Survey issued from a website well-known among the data science community, and the results of our google research build the base of our evaluation.
1) KDnuggets Usage Survey: KDnuggets is one of the most prominent website platforms dedicated to artificial intelligence and data analytics [4]. This platform usually conducts an annual survey to determine the usage rate of existing ML/DL tools. Currently, the 20th edition has been published. For example, it evaluates the usage rate of eleven different deep learning tools. With more than 1800

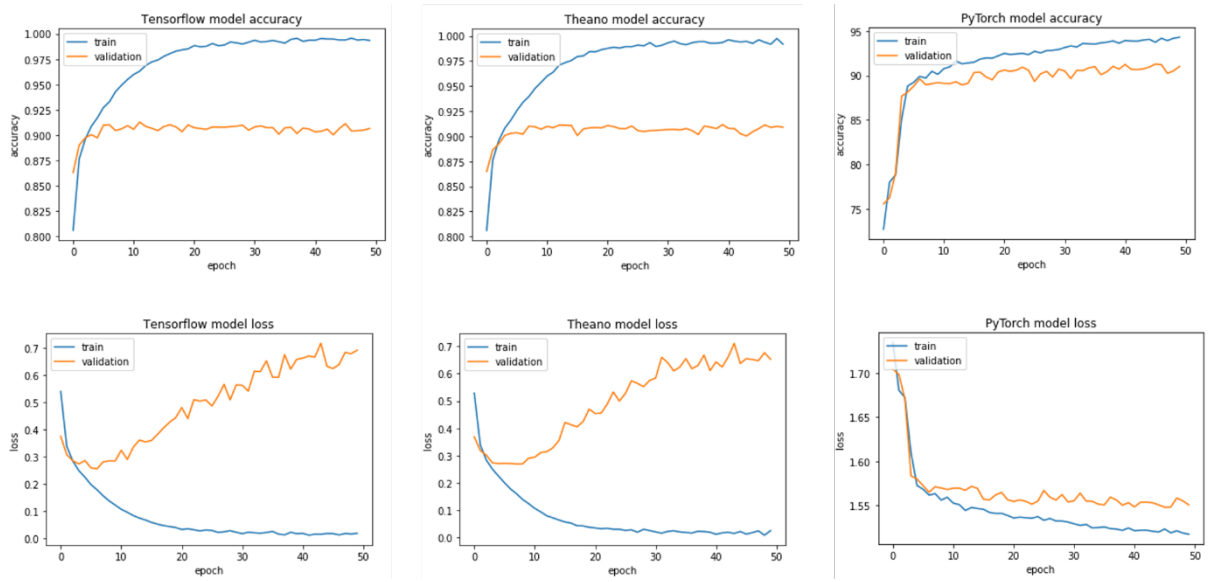

Fig. 2. Models Losses and Accuracies

DL Frameworks	Datasets						Training Time
	Train Set		Validation Set		Test Set		
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	
Tensorflow	**99.34%**	**0.019**	90.65%	0.69	91.21%	0.66	183.11
Theano	99.18%	0.025	90.90%	**0.65**	91.06%	**0.59**	**140.34**
PyTorch	94.31%	1.52	**91.01%**	1.55	**91.34%**	1.55	184.03

TABLE I
OVERVIEW DL FRAMEWORKS INTERNAL CRITERIA

participants, including machine learning engineers, data scientists, data analyts, etc., the results show a continuing trend, which is illustrated in the table below [5].

We note in table number II the three platforms chosen for our evaluation. As in the previous year, Tensorflow remains the most popular platform among users in 2019. PyTorch is the third most popular platform and shows the highest increase among all other platforms. In just one year its popularity among users has almost doubled. Further on, Theano is ranked 8th in this table. In contrast to its predecessors, its popularity from 2018 to 2019 has decreased considerably.

2) Google search activity: We also used the Google Trend results as another criterion to evaluate the interest of the DL frameworks we selected. It is a search tool that can be used to create reports, as it includes various categories and filters that can be used to narrow the results of a search. We entered the three frameworks as keywords and chose the category ”machine learning and artificial intelligence” to filter the results according to the requirements of our study. The figure below is a graphic that illustrates the results of our search.

The given graph is composed of our three selected ML/DL frameworks and shows the evolution of their popularity in the history of Google search over the years. While Theano (marked in red) already released in 2007, could have had an advantage over Pytorch, (marked in yellow) and TensorFlow (marked in blue), the graph shows that this platform has never really been very successful within the ML community. On the other hand, when Tensorflow was launched in late 2015, it was immediately strongly adopted by the community and its popularity has rarely declined until recently. In the same vein, Pytorch also seems to share a huge interest in the web community since its release in September 2016 and continues to grow since then. Moreover, we notice that even if Tensorflow remains the trendiest platform, its popularity has considerably decreased with the advent and sudden interest in PyTorch.

V. CONCLUSION

Our study focused mainly on image classification. We evaluated some of the best known ML/DL frameworks, including Tensorflow, Theano and Pytorch. The evaluation was based on several criteria divided into internal and

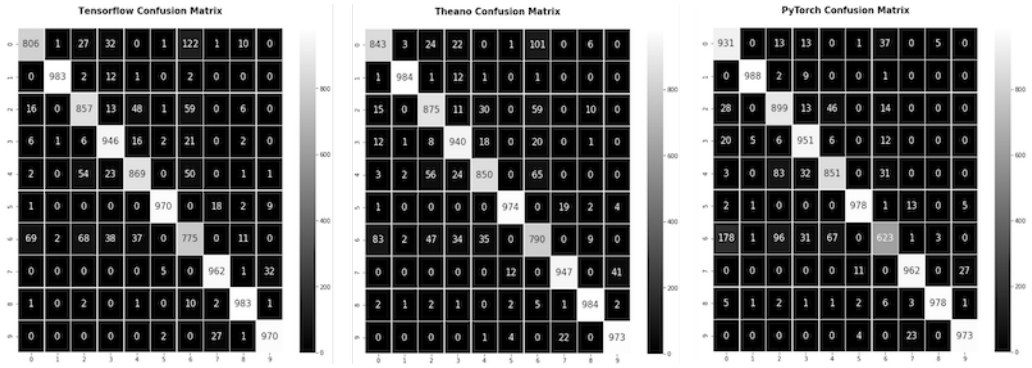

Fig. 3. Confusion Matrixes

Platform	2019 % share	2018 % share	% change
Tensorflow	**31.7%**	**29.9%**	5.8%
Keras	26.6%	22.2%	19.7%
Pytorch	11.3%	6.4%	**75.5%**
Other Deep Learning Tools	5.6%	4.9%	15.2%
DeepLearning4J	2.5%	3.4%	-25.6%
Apache MXnet	1.7%	1.5%	13.1%
Microsoft Cognitive Toolkit	1.6%	3.0%	-45.5%
Theano	1.6%	4.9%	-67.4%
Torch	0.9%	1.0%	-6.1%
TFLearn	0.7%	1.1%	-34.7%
Caffe	0.6%	1.5%	-58.3%

TABLE II
DEEP LEARNING PLATFORMS USAGE RATE [5]

Fig. 4. Frameworks Google Trend over years

external criteria. Firstly, internal criteria such as model loss and accuracy, training time and confusion matrix were used to evaluate the performance of each framework. Our results showed that all three frameworks achieved significant and similar levels of accuracy of approximately 91%.
Subsequently, with respect to the model loss, Tensorflow and Theano outperformed PyTorch which, unlike the other two, achieved a loss value greater than 0. However, it should be noted that the latter framework was also the only one that did not show an over-fitting on the validation set. As discussed earlier in this paper, the problem of overfitting was one of the shortcomings we encountered in the course of our study. It is obvious that this could have been avoided by first regularising the data and/or adding dropout layers to the models. Nevertheless, since we had to have a loss function that would be valid for all three frameworks, it required some changes, such as the removal of dropout layers, which accordingly led to that shortcoming. In addition, another metric used in our evaluation was the training time. Tensorflow and Pytorch were quite fast, with 183 and 184 seconds respectively. Theano was just behind the duo with a delay of almost 40 seconds. Secondly, we also used external criteria such as the level of interest and popularity of each framework to evaluate them. The KDnuggets Usage Survey and our Google Trend Search showed that Tensorflow is the most trendy frame in the machine learning community.

However, we observed that this trend is gradually shifting towards Pytorch. On the other hand, Theano has never really been adopted by users and interest continues to decline. Furthermore, we suggest ML researchers and practitioners to use this study as a starting point for their own research or work. Indeed, the choice of the appropriate framework is quite personal and may vary from one individual to another depending on the weight given to each criterion. Future researches may also focus on identifying the criteria that may be missing from this study. We also encourage trying to use the Sparse Categorical Crossentropy as a loss function. It is with the Categorical CrossEntropy we used in this evaluation, also best suited to multiclassification problems. In addition, it might also be very interesting to try other neural network models to see if the results obtained in this study vary significantly.

REFERENCES

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi. "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[2] Kai Dinghofer and Frank Hartung. "Analysis of Criteria for the Selection of Machine Learning Frameworks". In: *2020 International Conference on Computing, Networking and Communications (ICNC)*. 2020, pp. 373–377. DOI: 10.1109/ICNC47757.2020.9049650.

[3] Migran N Gevorkyan et al. "Review and comparative analysis of machine learning libraries for machine learning". In: *Discrete and Continuous Models and Applied Computational Science* 27.4 (2019), pp. 305–315.

[4] KDnuggets. *About KDnuggets*. URL: https://www.kdnuggets.com/about/index.html.

[5] KDnuggets. *Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis*. 2019. URL: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html.

[6] Chance Simmons and Mark A Holliday. "A comparison of two popular machine learning frameworks". In: *Journal of Computing Sciences in Colleges* 35.4 (2019), pp. 20–25.

[7] Igor Stančin and Alan Jović. "An overview and comparison of free Python libraries for data mining and big data analysis". In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2019, pp. 977–982.

[8] R. Vinayakumar, K. P. Soman, and P. Poornachandran. "Applying convolutional neural network for network intrusion detection". In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2017, pp. 1222–1228. DOI: 10.1109/ICACCI.2017.8126009.