Caffe2 vs. TensorFlow: Which is a Better Deep Learning Framework?

BAIGE LIU, Stanford University
XIAOXUE ZANG, Stanford University

Deep learning framework is an indispensable assistant for researchers doing deep learning projects and it has greatly contributed to the rapid development of this field. Among the great amount of the public frameworks, we focus on Tensorflow and Caffe2 and implement a detailed comparison of the two in the aspect of the expressiveness, modeling capabilities, help&support, performance on GPU, and the scalability. They both use static computational graph and is similar in structure, but Caffe2 is more speedy and takes less space. According to our experiments' results, the time of training using Caffe2 is about 2/3 of the time of training on Tensorflow and Caffe2 takes about 40% less memory space than Tensorflow. In terms of the performance on doing inference tasks, Caffe2's superiority is more conspicuous. Caffe2 takes about 75% less time than Tensorflow on the same inference task. However, Tensorflow has better help&support, provides more services and functions, and is thus a better choice to realize novel and complicated models.

1 INTRODUCTION

As deep learning is becoming a hot topic, more and more researchers and engineers are using deep learning techniques to help solve big data problems such as computer vision, speech recognition, and natural language processing. There are many benefits to use a deep learning frameworks. For example, we can easily build big computational graphs, compute gradients in it and run it all efficiently on GPUs(wrap cuDNN, cuBLAS, etc)[7]. Researchers are free from the pain of starting everything from scratch and rebuild things that may have already been implemented by other people before with just a slight difference. Also, it helps to form a community where everyone shares the common grammars for writing programs so that they can easily communicate with each other and it enables more code reusability. Since a good deep learning framework accelerates the progress of the work and is essential for doing successful deep learning projects, choosing the appropriate framework suitable for the task is an important next step. Our work intends to give people guidance in making their choices from the various kinds of the existing frameworks. We believe there is no machine learning framework that can beat any other frameworks. There are always some advantages and disadvantages to use a particular deep learning frameworks. Therefore, we compare the frameworks in as many aspects as we can think of to give the reader a thorough understanding of the frameworks so that they can choose which one to use more reasonably according to their own project needs. We focus on two widely-known deep learning frameworks: Caffe2 and Tensorflow and make a detailed comparison in five aspects: the expressiveness, the modeling capability, the performance, help & support, and the scalability. We choose Tensorflow because it is currently the most widely-used deep learning framework. Caffe was an extremely popular framework before Tensorflow was introduced and we believe there is a large potential that the new framework Caffe2 will gain a lot of user preference in the near future. We find Caffe2 and Tensorflow do not differ so much in expressiveness, the modeling capability, and the scalability, but Caffe2 significantly performs better than Tensorflow in both speed and space aspects. Therefore, Caffe2 is a better choice for people who pursue speed or are limited by the device restrictions. On the other hand, Tensorflow provides more services and tools, such as Tensorboard, Tensorflow serving, Tensorflow Lite, and has a strong advantages in help&support, it is a better choice if people want to implement new or complicated models and do not know how to implement exactly yet.

We first briefly introduce the background and the similarities of the two in section 2 and then elaborate on their differences in the following section 3 - section 7. Finally, we conclude our comparison and illustrate the possible future work in Section 8.

2 SIMILARITIES

Caffe2 and Tensorflow are respectively developed by Facebook and Google, which have different strategies in building their frameworks. Facebook describes Caffe2 as "a lightweight and modular deep learning framework emphasizing portability while maintaining scalability and performance.", which in short sentence, emphasized the advantage of Caffe2 in production(Facebook uses Pytorch as research purpose). In contrast, Google aims to build Tensorflow into a all-in-one solution for various machine learning tasks and thus, Tensorflow is more complicated, comprehensive, and bulky. They share the similarities in the following points:

- They supports auto gradient computation, both use C++ for computing, and provide Python and C++ interface,
- They allow users to deploy trained models on a variety of servers or mobile devices without having to implement a separate model decoder or load a Python interpreter.
- Both support auto gradient computation, use the static computation graph, and precompile the network to obtain the optimal training performance.
- They support multiple machine distributed computing.
- They support deployment on CPU, NVIDIA GPU hardware and the deployment on mobile systems.
- They are open sourced.

3 EXPRESSIVENESS

We compared the expressiveness of Caffe2 and Tensorflow by two metrics: easiness to write, easiness to debug. Finally, we also implemented a simple user study to investigate the true using experience of these two deep learning frameworks.

3.1 Easiness to write

3.1.1 code style. Tensorflow and Caffe2 both use static computational graphs to achieve better performance and thus are very similar in their code style. Tensorflow uses Tensor as their data units. Every node including operators in the computation graph is a basic Tensor. In contrast, Caffe2 uses Blobs to define the container of data and a seperate concept - Operators to define the methods. Operators take Blobs as the input and the output and do the computation inside. Operators are actually realized as protobuf objects. In Caffe2, Net is the computation graph and defines the architecture of the network as protobuf object. Workspace is where Net and all the variables reside. Workspace is a similar concept to Session in Tensorflow. However, the difference between them is that Workspace initializes themselves the moment it is used. In contrast, Session needs to be explicitly initialized before using it to run the graph. The relation of Caffe2 and Tensorflow is summarized in Table 1.

Table 1. The comparison of Caffe2 and Tensorflow in concepts.

	Tensorflow	Caffe2
Data Unit	Tensor	Blob (a typed pointer that can store any type of C++ objects)
Operator	Tensor	Operator (protobuf object)
Network scope	tf.session()	Workspace
Graph	tf.graph()	Net (protobuf object)

3.1.2 Code flow. Tensorflow code follow the flow of Graph definition → Session (variables) initialization → Session run. In contrast, Caffe2 code is composed of only two steps: Graph definition → Workspace run. We compared the code written in Tensorflow and Caffe2 for handwriting recognition task and found the code structure quite similar [1][2], thus we consider two are of the same level in the aspect of the easiness to write if the user builds their own models. However, since Caffe2 is using protobuf object in its design, it enables people to define the network only by editing the prototxt and to train the model by running a simple script. Model Zoo is a useful resource of pre-trained models. It makes it easy to do fine-tuning tasks on the well-known neural networks or to do inference tasks using pre-trained models in Caffe2. But Caffe2 has its disadvantages. If the network is complex and of big-scale like Residual Network (ResNet) or GoogleNet, the prototxt file becomes tedious and complicated. Thus, we think it is easier to write the code in Caffe2 than Tensorflow if the machine learning task only utilizes the pre-trained model and is not complicated. In

3.2 Easiness to debug

First, Caffe2 is much harder to install than Tensorflow for the large amount of the required dependencies. In contrast, Tensorflow installation can be done in only one line of pip install command. Caffe2's complicated installation procedure increases the debugging time and may daunt users away from further uses.
However, because Caffe2 uses protobuf objects to represent network architecture, it is easier to print out the architecture and visualize the graph in place solely by a net_drawer() internal function. On the other hand, to visualize the graph in Tensorflow, one needs to use Tensorboard. Though Tensorboard is a powerful tool supporting not only the visualization of the graph, but also the monitor of the training process (automatic plot of loss and metrics) by tf.summary module, learning its usage and embedding them into the code takes some time. It is especially hard for machine learning and Tensorflow beginners. Thus, we reckon that it is easier for people to pick up Caffe 2 than Tensorflow and to debug for simple deep learning models. However, if the neural network is complex and the project is research-oriented, Tensorflow supports more flexibility and Tensorboard would show its advantages in debugging.

3.3 User experience

In order to evaluate how the two frameworks are easy to be used, besides comparing their documentations, the code flows etc, we also conducted a user study by asking eight users who have experience in deep learning to answer some questions in the user experience survey. The survey basically has the contents including the time it takes the users to install the framework etc. The detailed questions can be found in this Google Questionaire [1]. Although this is just a coarse user study as our sample size is small so there is a non-neglectable bias, the findings are still interesting to present. First, 75% of the users take less than twenty minutes to install Tensorflow on their own PC. However, on average it takes much more time to install Caffe2 both on local machine and remote server. Second, the average score the users give for the Tensorflow documentation is 4.0 compared with 3.375 for the Caffe2 documentation. However, since Caffe2 is a relatively new framework, we expect Caffe2 to improve its documentation after gathering more user feedbacks in the future. Third, 75% users would prefer to use Tensorflow for building their CNN models and

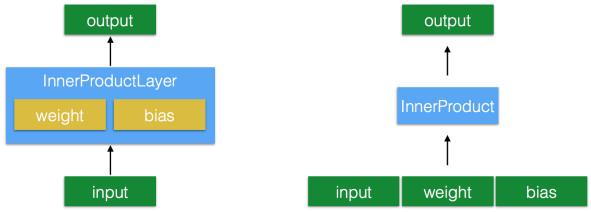

Fig. 1. Operator functionality comparison.[6]

87.5% users prefer Tensorflow for building their RNN models.

4 MODELING CAPABILITY

Whether users can use the framework to create their own instances is an important factor evaluating the capability of the framework. One of the big improvements of Caffe2 from Caffe is its finer granularity. As Figure 1 shows, previously network is constructed by the unit layer and if users want to build their own customized layers, they need to write functions illustrating how gradient is computed. However, Caffe2 replaces the concepts of layers with operators. It increases the flexibility of the networks and makes it easier to build customized networks. Caffe2 supports 400 operators and users can define their own operators by writing C++ code with details of the usage, input, output, and how gradient is passed. Tensorflow has the same concept and support customizing operators in the same way. Both Caffe2 and Tensorflow offer default python wrappers and general unit test functions for tests. Although there might be some minor differences in how to the operators are defined, overall, we consider in the terms of modeling capability, Caffe2 and Tensorflow are of the same level.

5 PERFORMANCE EVALUATION

We compared the computation efficiency between Caffe2 and Tensorflow by doing training and inference tasks using CIFAR-10 dataset. CIFAR-10 dataset is consisted of 60,000 32x32 color images in 10 classes, with 6,000 images per class. We compared two deep learning architectures of different scales: VGG-style CNN model with 6 layers and VGG-16 CNN with 16 layers. We used NVidia Tesla K80 GPU for all the experiments. We report the performance by comparing their execution time

Table 2. The training time of using Caffe2 and Tensorflow to train small CNN model on CIFAR-10 dataset.

	Tensorflow	Caffe2
Graph construction and initialization	4.71s	1.89s
Training	89.43s	55.71s
Inference	3.96s	0.40s

and memory space they took. We ensure that our implementations are correct by achieving the accuracies that are claimed by the original papers using those models.

5.1 Speed

First, we used the simple VGG-style network of 6 layers to test the performance. We conducted training for 10 epochs with batch size as 64 on 49,984 images. We took the average of 10 runs and the result is displayed in Table 2. The inference was tested on 9,984 photos and was conducted for 10 times as well. We took the average and report the time in Table 2. We compare the time of graph construction and initialization, training and inference. For all three parts, Caffe2 cost significantly less time than Tensorflow. The time Caffe2 costs on training is only 62% of the time Tensorflow costs on trining and the time Caffe2 costs on inference is only 10% of the time Tensorflow costs on inference. However, we want to exclude the possibility that the reason Tensorflow performs worse is that the model lacks of complexity and Tensorflow models will eventually be better than Caffe2 models when the neural network is very large. Therefore, we evaluated the speed performance of the two frameworks with increasing number of layers using VGG11, VGG13, VGG16, VGG19 network architectures. We plot the result in Figure2. We can see from the plot that the training time is roughly linear to the number of layers we have and it is explainable because the layers in the network are by nature similar for the same VGG architecture. And for the same model structure, the model implemented using TensorFlow generally takes more training time than that implemented using Caffe2 despite of layer number difference. Therefore, we expect Caffe2 to constantly perform better than Tensorflow with regard to speed no matter how large our model is. We consider that the worse performance of Tensorflow might result from how tensor object is implemented in Tensorflow. We speculate tensor object might have

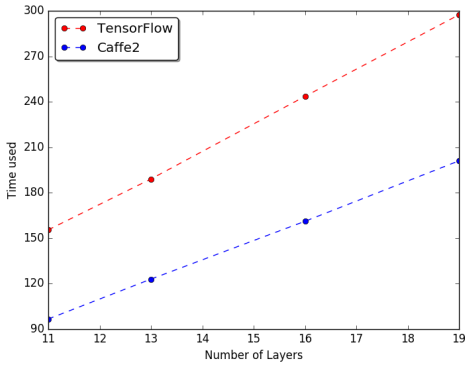

Fig. 2. Training Time for different network architectures

Table 3. The memory space of using Caffe2 and Tensorflow to train small CNN model and do inference using the trained model on CIFAR-10 dataset.

	Tensorflow	Caffe2
Training	3.960GB	2.424GB
Inference	3.405GB	1.854GB

bigger overhead and doing calculation with tensors is time-consuming. To justify our speculation, we computed the time of doing tf.argmax(), which is to compute the class with the highest likelihood after the softmax layer in the inference stage, and we compared it with the time of using numpy.argmax() to do the same thing in Caffe2. The time of doing tf.argmax() on tensors of shape (10, 1) for 9984 times was 0.4066s. In contrast, the time of doing numpy.argmax() on numpy array of shape (10, 1) was 0.0075s. The big gap supports our hypothesis that doing calculation on tensors cost more time. Although further experiments and comparison is needed to fully justify our speculation, the slow computation speed of tensors is one of reasons for tensorflow's longer training and testing time than Caffe2.

5.2 Space

We recorded the space it took to do the training and inference tasks using VGG-style 6 layer neural network on the 49,984 training dataset and the 9,984 test dataset. The result is presented in Table 3. Caffe2 took almost half the space it took for Tensorflow.

We also run the VGG7, VGG11, VGG13, VGG16, VGG19 models on GPU using the two frameworks. Generally, we consider the framework to be better in terms of the

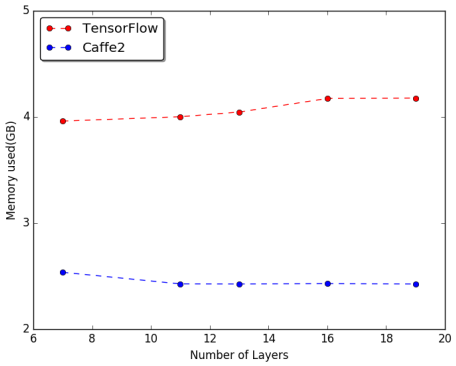

Fig. 3. Memory used for different network architectures

space complexity if it takes less memory. As we can see from Figure3, Caffe2 occupies less memory than Tensorflow regardless of the model complexity.

6 SCALABILITY

Scalability is the concept of running parallel jobs across multiple machines in a distributed system. One way to achieve the parallelism for faster training and testing in deep learning tasks is to parallel the model by splitting it into different portions so that they can be trained on different devices in parallel.

According to the tensorflow official documentation [4], the users may set up different tasks on different machine(including different GPU devices on the same machine) and associate each task with one Tensorflow server. The Tensorflow cluster consists of all the tasks for the execution of a TF graph. Inside each task, there is a master that creates the session and a worker that executes operations.

To enable the model parallelism on multiple machines, tensorflow uses the statement "with tf.device(...)" to specify which part of the model to be run on a particular machine. In addition, it keeps a `tf.train.ClusterSpec` which is shared among all the devices and a `tf.train.Server` instance kept in each task to ensure the smooth communication between the machines. However, there is much information passing such as activation values and gradient values between different layers in the deep neural network acrossing the distributed system if we are consering the model parallelism. Another parallelism technique is called data parallelism also named as repliated training where the machines in the distributed system train the same model with different mini-batch of the data. The important issue is to ensure the shared parameters are updated correctly among the machines. Tensorflow gives several possible approaches including in-graph replication, between-graph replication, asynchronous training an synchronous training.[4]

In Caffe2, the scalability has been designed to be an important feature and it is well-known for he multi-GPU acceleration. The data parallelism is a built-in library so users only need to write code to realize model parallelism. In addition, Caffe2 features built-in distributed training using the NCCL multi-GPU communications library which means that you can very quickly scale up or down without refactoring your design, while Tensorflow requires the users to define their own. For example, in Caffe2, most of the built-in functions seamlessly toggle between CPU-mode and GPU-mode depending on where they are running.[2] In addition, Tensorflow is relatively harder to optimize in terms of the scalability because of its granularity.

Caffe2 claims that it achieves close to linear scaling with Resnet-50 model training on up to 64 NVIDIA Tesla P100 GPU accelerators (57x speedup on 64 GPUs vs. 1 GPU)[2]. However, the actual performance comparision between the two frameworks in distributed system will be included in future works as we currently do not have enough GPU computing resources to run experiments on.

7 HELP & SUPPORT

Although Caffe2 and Tensorflow have python and C++ APIs, Caffe2 only supports python2, while Tensorflow supports both python2 and python3. Besides the language they support, we compared the available resources and the hardware they can be deployed on as follows.

7.1 Resource

Since Caffe2 is a relatively new framework, the community size is smaller than that of Tensorflow and thus less online code resources. According to the recent study at the beginning of May of 2017 that summarises the frameworks used by most of the popular open source deep network repositories in Github, we can see there are roughly ten times more Tensorflow users than Caffe2 users.[3] Also, according to the statistics of Stackoverflow posts related to the deep learning frameworks, TensorFlow is clearly leading the race for deep learning framework adoption.

Tensorflow has many high-level wrappers like Keras, TFLearn and TensorLayer which makes common things easy to implement because coding static graphs is generally thought as not as intuitive and as coding dynamic graphs. Caffe2 has its advantages in the existing pre-trained models available in Model Zoo and Caff e2 team provides the public translator tool to easily translate Caffe's model to Caffe2's model, which is very helpful and supportive because there are many ongoing projects in industry using Caffe models. However, translating the Caffe model to Caffe2 model usually cost hours. Overall, Tensorflow is more supported and have more online resources than Caffe2.

7.2 Hardware support

Besides supporting generic CPU and NVIDIA GPU, Caffe2 is officially claimed to be suited for the deployment on mobile devices and work within the low-power constraints of such devices. For example, caffe2 is used by Facebook for fast style transfer on their mobile app. It is said to "harnesses the power of Adreno graphics processing units and Hexagon digital signal processors on Qualcomm Inc.'s Snapdragon chips" to achieve this goal. We did not implemented experiments to confirm Caffe2's actual performances, but Caffe2 library's package size is only 37.1MB, nearly one third of that of the Tensorflow library. Tensorflow has recently published Tensorflow Lite to complement its shortage in mobile and embedded devices. The architecture of Tensorflow Lite is described in Figure 4 and since it is still under development, currently it only supports limited operators. Future work is required to do the comparison of both in mobile devices.

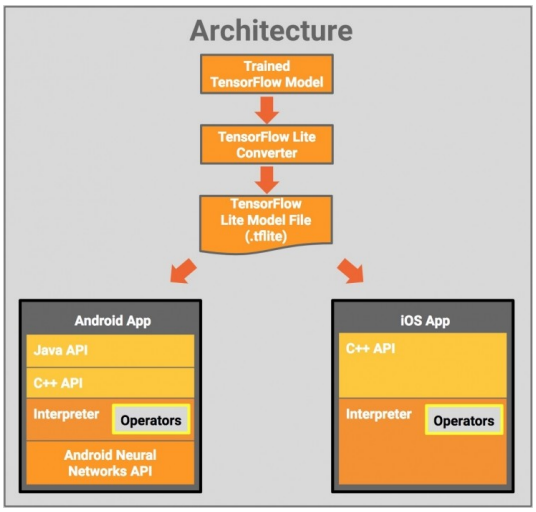

Fig. 4. Tensorflow lite architecture[5]

8 CONCLUSION

We compare Caffe2 and Tensorflow in many aspects and as a result we find neither of these two has an dominating advantages over the other. Therefore, in pratice, the choice between these two actually depends on the specific user tasks and the user preferences. Overall, if the user need to pursue speed and has limited space restricted by the device, Caffe2 is a better choice since our experiments' results revealed that Caffe2 has a significant advantage over Tensorflow both in speed and space. Nevertheless, Tensorflow is still powerful and useful because there is a large number official and third-party resources, services, debugging tools, and a big supportive community that makes it easier to find reference codes. Thus, we think Tensorflow is a better framework for implementing a complicated or innovative networks compared to Caffe2.

A LIST OF WORK

Xiaoxue Zang: Set up experimental environment. Wrote the Caffe2 code for the experiments and run the experiments. Compared the expressiveness, modeling capability, and help & support.
Baige Liu: Implemented user study and conducted analysis, Wrote the tensorflow code. Compared the scalability, similarity, supplemented additional materials to other parts.

REFERENCES

[1] Xiaoxue Zang Baige Liu. 2017. Questionaire. (2017). https://docs.google.com/forms/d/e/1FAIpQLSfjyq0U-i4uuc9Wf16SaojoinhiDZ5v_wi6wds25Y1Vwt3fmg/viewform?c=0&w=1

[2] Nvidia blog. 2017. Caffe2: Portable High-Performance Deep Learning Framework from Facebook. (2017). https://devblogs.nvidia.com/parallelforall/caffe2-deep-learning-framework-facebook/

[3] Aymeric Damien. 2017. Deep learning frameworks. (2017). https://www.cio.com/article/3193689/artificial-intelligence/which-deep-learning-network-is-best-for-you.html

[4] Google. 2017. Introduction to Distributed Tensorflow. (2017). https://www.tensorflow.org/deploy/distributed

[5] Google. 2017. Introduction to TensorFlow Lite. (2017). https://www.tensorflow.org/mobile/tflite/
[6] Yangqing Jia. 2015. Improving Caffe: Some Refactoring. (2015). http://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-improving.pdf
[7] Feifei Li. 2017. Deep Learning Software. (2017). http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture8.pdf