1. What is the key need for this algorithm/approach/system? What are some short comings of existing approaches?

Scientists should write their function all by themselves before, such as gradient descend, neural network, activation functions, etc. This is really time consuming for the engineers to develop them all, and might make some mistakes that is very hard to find. Nevertheless, TensorFlow is a library or framework and it can help scientists develop their own machine learning model in just few lines. Hence, even people who are beginners in machine learning area can also build their own model in a short time.

2. What is the key object (term) that is the solution to this need? (e.g. tweet, RDD, Tensor) Describe this object in a few sentences. This is the fundamental concept or "thing" proposed to solve the need addressed in question #1.

TensorFlow is described by a directed dataflow graph and values flow through edges in graph are tensors (it is presented by a typed, multidimensional array). When initializing a new session, it is empty. We put variables, loss functions, activation function, optimizers into our session(graph). By doing so, we can build our model, and then call session.run to start training the parameters. All these functions are built in, so users can get rid of the difficult math problems when creating models. Therefore, TensorFlow simplifies the process of machine learning.

3. What has the author identified as a weakness or limitation of the proposed algorithm / approach / system? Or what has the author proposed as next steps? If the author does not provide this information, what do you think could be improved?

The authors mentioned that performance of TensorFlow showed that there is still room for improvement. There are several methods to improve it. For example, placement and node scheduling strategies that are used to decide when and where each node will execute can become more efficient by a system that can learn how to make good placement decisions. Moreover, TensorFlow only supports Nvidia GPUs, this is really bad for AMD users.

4. What is something interesting your learned from this paper, or your thoughts about its strengths and/or weaknesses. Is there anything else interesting about this paper, or your interpretation of it that you want to share?

TensorBoard is a great feature because it can help us understand what the process of our model by graph. In addition, it can also show how the state changes during the training time, such as gradient, weight or bias. This really help users evaluate the models more precisely since we can track what is happening during training period. Moreover, the visualization of the graph enables users to check whether the model is what we would like to implement or not. Thus, we can get more information from TensorBoard but not just from the code we write.

5. Read an additional related paper. Provide a citation for this paper. How does this paper relate to the assigned one? What new information did you learn by reading it? This answer should be a decent sized paragraph describing the content of this paper (be specific) and how it relates to the assigned paper.

Rethinking the Inception Architecture for Computer Vision

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. arXiv preprint, 1512.00567, 2015.

In this paper, the authors provided several design principles to scale up convolutional network. It teaches me how to build a CNN to do image classification by using TensorFlow on ImageNet. For example, what filter size should I set to get higher accuracy, how to do padding and create multiple layers in the CNN model. This is really helpful after reading the introduction of TensorFlow (previous paper) since I can learn further by a practical example that presents in the paper. Moreover, the author introduces the inception that has multiple sizes of filters on the same level and it solves the several problems. For instance, the very deep CNN might cause overfitting, and the gradient would very hard to pass back when do the back propagation. Hence, inception can get our model wider instead of deeper, and it also can make computational expenses cheaper. It can be done by using TensorFlow and that is elegant.