CMPUT 466 Mini Project Final Draft

Name: Minghan Li Student ID: 1561234

1 Abstract

Deep neural network has achieved tremendous success in areas such as image classification and speech recognition. However, recent studies suggest that a shallow neural network would also be able to perform nearly as good as a deep model on complex datasets if it is learning from a teacher model. This method is known as model compression[1] or distillation[2], which uses a trained model that has relatively high accuracy to label the dataset, and then trains the shallow model with these "fake" data. Although model compression or distillation has already shown that the performance of shallow models could be improved by the extra information given by the teacher model, it is still questionable whether we form this teacher-student learning paradigm properly, as well as whether the performance of the student model can be further improved. In this mini-project we are going to compare three teacher-student learning methods on the CIFAR-10 dataset[3] and perform an empirical evaluation on those results. The first two are the methods described in [1] and [2], and the last one which is proposed by myself is called multi-channel label method. The nature of this multi-channel method is converting the multi-class classification problem into a multi-label classification problem, which makes the teacher model produce an embedding-like prediction that can help the student model better discriminate the data. We would argue that this transformation actually forms a more appropriate teacher-student learning paradigm and the results in this experiment indeed shows that the student model using the multi-channel method has better perforamance than the other two methods.

2 Dataset and other tools

Dataset The CIFAR-10 dataset[3] consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. All the images have been preprocessed: the object is aligned at the center and the background is clean, which makes it an ideal testbed for new algorithms in computer vision and machine learning; however, the complexity in real life objects can still be very challenging, thus the results are more persuasive than the one obtained from some "simpler" datasets, e.g. MNIST[10].

Programming langugage and packages The programming language is Python. Numpy[7], Scipy[8] and Tensorflow[9] are our main tools for constructing models and performing statistical tests. All the models in this project are implemented mainly in tensorflow and numpy by myself.

3 Detailed explanations

The motivation for this project is to figure out why using the predictions of the teacher model instead of the original labels can help the student model to learn better. Several hypotheses have been proposed[1][2]. According to [2], the predictions actually encode some similarity metrics between classes and thus help the student model to better discriminate different samples in the same class, as well as recognize the relevance between different classes. Figure 1 shows the semantic meaning of teacher's prediction.

However, subsituting the original labels with the teacher model's predictions and then directly minimizing the cross-entropy directly might give problematic results. Therefore, two schemes have been proposed to help the shallow student model to learn better. The first one proposed by [1] is to directly minimize the L2 loss between the teacher's logits and the student's logits ("logits" refers to the ouput before the softmax activation), which I will refer to it as "logits learning" in the rest of this report.

The second one is a technique called "distillation" which is proposed by [2]. They actually propose to minimize the cross-entropy loss between the student's softmax predictions and the teacher's predictions, but both predictions of the teacher and student are "softened" by "high-temperature". We won't discuss the details here but the nature of it is to approximate minimizing the L2 loss between

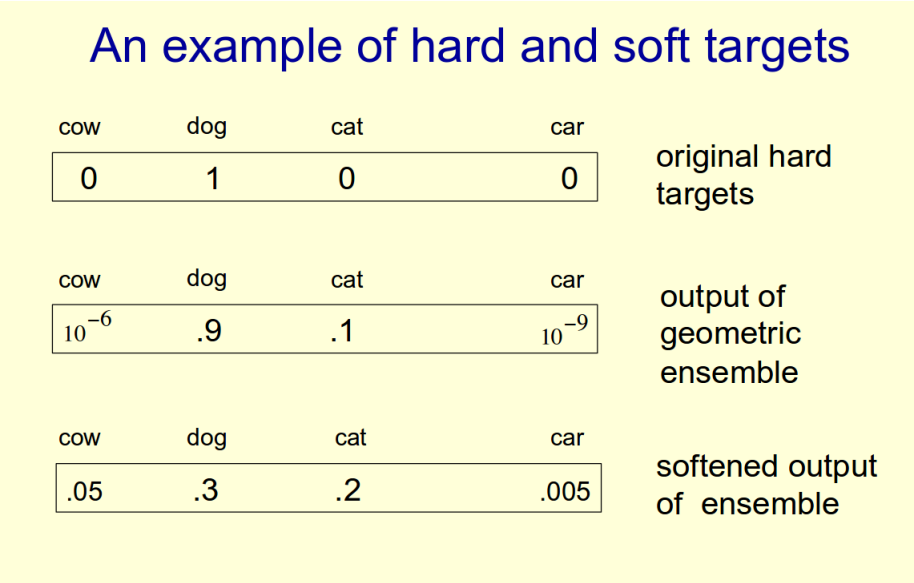


Figure 1: An example of the teacher's prediction and distillation from Geoffrey Hinton's ppt "Dark Knowledge"[4].

the logits mentioned in the first method. This method has another benefit which allows us to mix the original labels and the teacher's predictions together, so that the student might learn better than simply mimicking the teacher.

Although both of these two methods work pretty well over many datasets and different teacher-student model pairs, it is still questionable whether it is theoratically valid to minimize the cross-entropy loss between the predictions and the fake labels such as [0.6 0.2 0.2] rather than [1 0 0]. Using categorical loss implies that the labels are random variables which come from a multinoulli distribution (or bernoulli distribution if it is binary classificaiton) given the data, so the labels can only be binary vectors. Consequently, labels like [0.6 0.2 0.2] are technically not valid to use. However, the distillation method works pretty well in practice so this doesn't seem to be a problem. But actually we can do better. After inspecting the distribution of the teacher's predictions carefully as well as conducting several exploratory experiments, I think there's another way to present the knowledge in the teacher model that can better fomulate the problem, which is called multi-channel label method as illustrated in Figure 2.

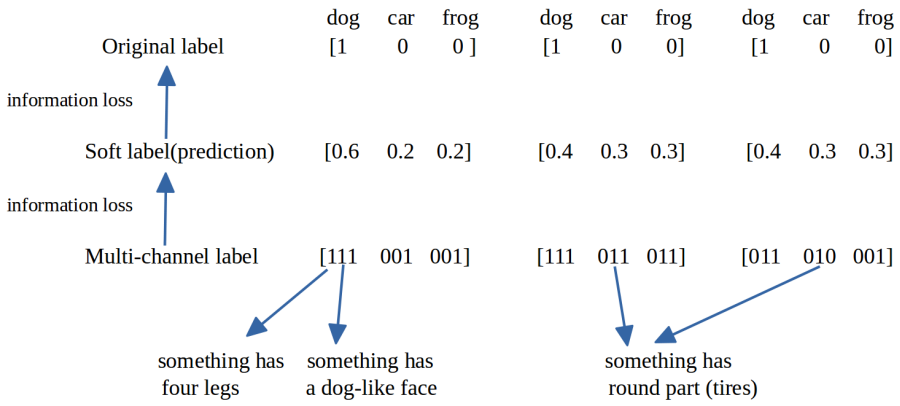


Figure 2: Semantic meaning of multi-channel labels and information loss

For example, if we look at the teacher's predictions that predict class 0 such as [0.6, 0.2, 0.2] and [0.6, 0.4, 0.0] carefully, we can find there's no way to represent them in 3-bit binary vector without loss of information; the best we can do is to represent them as [1 0 0]. However, if we use 3 "channels" to represent 1 bit, then we can represent them as [111 001 001] and [111 011 000] respectively. We don't have to make all the channels sum to 1, because the sum of every 3 bits determines which is the domiant class. Each channel is a bernoulli random variable: 1 means some structures of the class is presented in the given data, 0 is not. This turns the multi-class classification problem into a multi-label classification problem: whether some specific sub-structures of each class are presented in this data. As illustrated in Figure 2, the regular label is just a special case of multi-channel label, which is 1 channel label, and the multi-class classification problem can be viewed as a multi-label problem where every data point only has 1 label (or only one "1" in the one-hot encoding vector).

It is obvious that the more channels we have, the more information we can encode in one label. In addition, adding redundancy in the label might also in a way prevent overfitting (which is just a hypothesis). However, most categorical datasets are designed for the multi-class classification problem and the labels only have 1 channel. Fortunately, in the teacher-student learning paradigm, we can let the teacher produce multi-channel labels for the student with only small modifications to the original model. The modifications are illustrated in Figure 3.

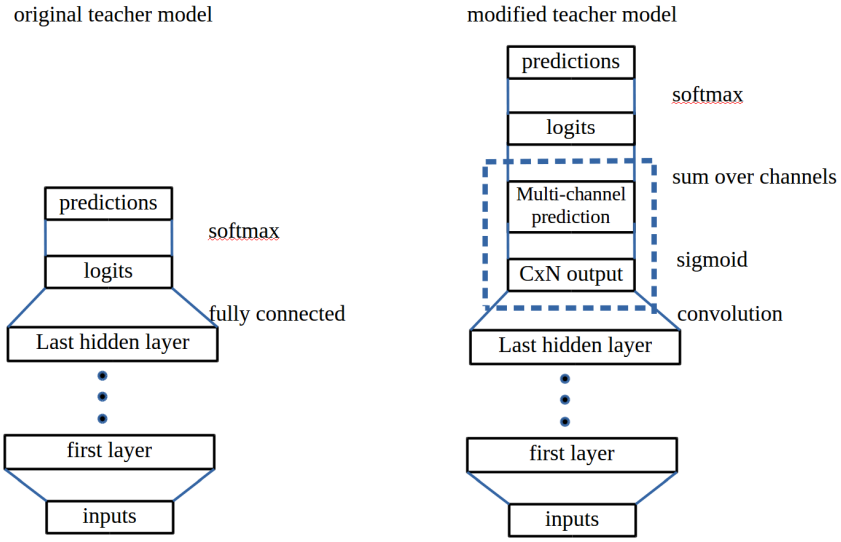


Figure 3: Modification on the teacher model for multi-channel label method.

Suppose we are using a neural network as a teacher model. As we can see, the structure of the teacher model will be exactly the same from input to the last hidden layer of the network. But instead of using the linear combination of the hidden representations to generate the logits, we use convolution with the sigmoid activation function to get the multi-channel predictions; then we sum up all the channels in a class to produce the logits and pass them to the softmax activation function. Finally we calculate the cross-entropy loss between the original label and the final predictions to train the teacher model. All these above are not aiming to improve the performance of the teacher model but to obtain the multi-channel labels.

It is not normal to use convolution on the last hidden layer. The reason to do this is because we want to produce many different local models which can be used to predicted whether some sub-structures exist in the data, and convolution happens to be able to extract local structures for each channel but meanwhile shares the weights over the channels in the same class so that the ensemble of the local models still predicts for one big class; the sigmoid activation is just to normalize the output and obtain the confidence of predicting whether one subclass exist; summing up the channels is to get the votes for the final class prediction, so that the class with the highest votes "wins". One might argue that by putting an extra sigmoid activation we change the teacher model. Indeed, the structure has been changed a little, but there's no weights to be learned after the sigmoid activation so it is not clear

whether the change in the structure will actually influence the final performance of the teacher model a lot.

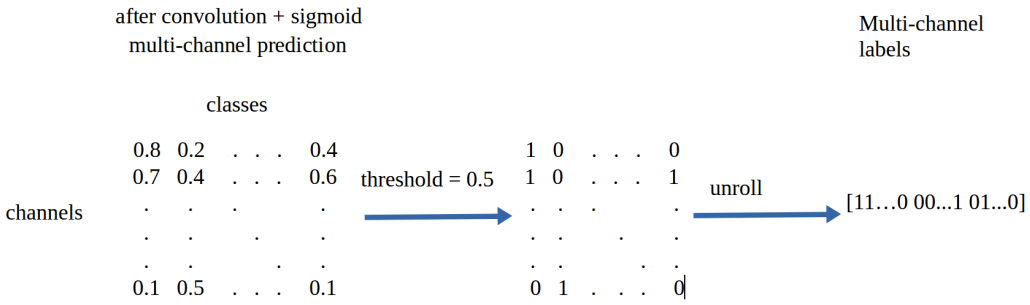


Figure 4: extract multi-channel labels from the teacher model

After the teacher model is trained, we first extract the multi-channel prediction as shown in Figure 2, and then thresholding the predictions to get the binary multi-channel labels(e.g. threshold = 0.5). Of couse, in order to train the student model with multi-channel labels, we also have to make a little modification on it, which are shown in Figure 5. Finally we minimize the cross-entropy loss between the multi-channel predictions of the student model and the multi-channel labels. Experiments indeed show that the performance of using the multi-channel model is slightly better than logits learing and distillation methods.

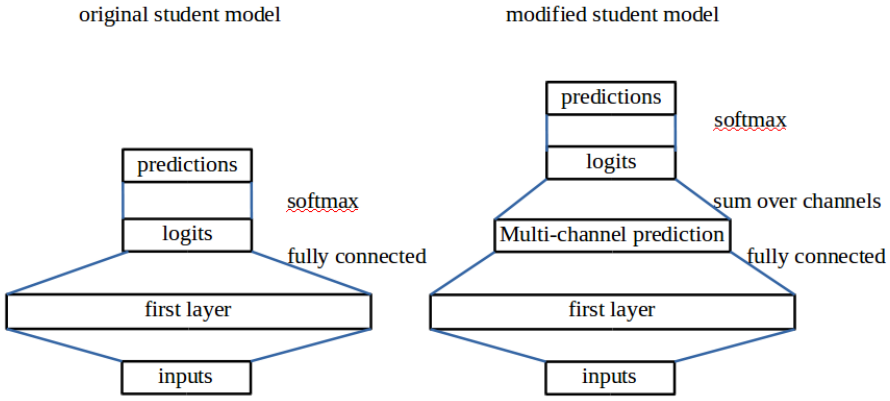


Figure 5: extract multi-channel labels from the teacher model

4 Experiments

Since we are comparing three different types of learning paradigm, the proper thing to do is to compare different teacher-student model pairs using these three paradigms. However, due to limited computational resources, we only choose one model to be the teacher and one model to be the student. The experiment specifications are stated as follows.

Teacher Model: We use network in network (NIN) model[5] as the teacher in this project. For the logits-learning method we use the original NIN model as shown in Figure 4; for distialltion method we set T=10 in the softmax activation function; and for the multi-channel label method, we use a convolution layer with sigmoid function between softmax activation and the last hidden layer instead of the original linear layer. The covolutional layer has 10 6x6 filters with stride 2 which will

produce a 36-channel output. We use dropout(keep_prob=0.5) to regularize the teacher models, and for comparison sake, we will train the teacher models to have almost the same accuracy on the test sets (which is also close to the highest accuracy among all three teacher models).

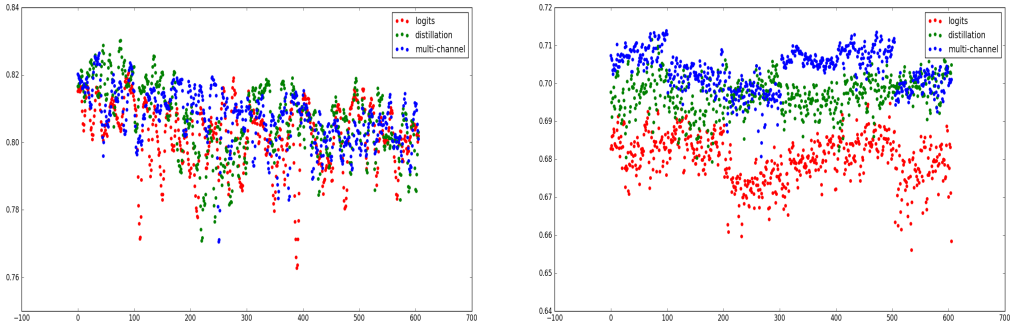
Student Model: We choose a shallow neural net with one convolutional layer and linear bottleneck to be the student as described in[1]. The convolutional layer has 128 3x3 filters with stride 1 and the size of linear bottleneck is 1200. The fully connected layer has 8x1024 neurons. The output size is 160 for multi-channel label method while training. No regularization method is used in the training of the student models. The structure of the student models are shown in Figure 5.

Training and Testing: In training we choose L2 loss for logits learning and cross-entropy loss for distillation as explained earlier. To optimize the loss function, all teacher and student models use mini-batch gradient descent method with batch size equals 128 as well as use Adam optimizer with 1e-2 as initial stepsize. Due to limited resources, the dataset is only divided into 6 subsets with equal balanced data and for each run we use a different subset as the test set and the other as training set. We will run 100 experiments on each test sets and each run lasts for 50 epochs(1 epoch means goes through the whole training set once) or more to make sure they reach the same test accuracy. Moreover, we set the baseline to be the student model learning directly from the raw labels without any teacher.

Hyperparameters Selection: Due to limited resources I can't afford to do k-fold cross-validation on so many "big" models, so the hyperparameters are picked mainly based on my empirical experience and some results represented in [1][2]. Also remember we trained the teacher models to the same accuracy on test sets to make a rather fair scenario, so tuning on the hyperparameters of the teacher models doesn't seem to be that important as long as we make the teachers have the same performance. However in multi-channel label method, the number of channels indeed might affect the student's performance, so we pick a rather moderate parameter setting which is 36 channels label in this experiment.

5 Empirical Evaluation

We first show the final performance of the teacher models and the student models on 6 different test sets over 600 runs:



(a) Teacher Models (b) Student Models

Figure 6: Test accuracy on 6 different test sets over 600 runs

We use accuracy instead of other metrics as measurements is simply because the dataset we use are very balanced and we have done many runs on each training and test set. Indeed we can use other measurements like F-score, but we just don't feel it is necessary because simple metric like accuracy is enough to reflect the performance in this case.

We refrain the test accuracy of all the three teacher models to be the same, which is around 81%. Some would argue that it might be better to train the teacher models to their best performance and

then compare the performance improvement on the student models . In this case although the best performance of the three teacher models are different, they are around 81% to 83% of accuracy, which shouldn't cause too much difference on the predictions than the optimal models, and it makes statistical significant test easier as well.

Test results of both teahcer models and student models are shown in Figure 6. For every model we test it on 6 different test sets (since we split up the whole dataset which also means we use 6 different training sets as well) over 600 runs (i.e. training eache model on each training set and then testing them on each test sets 100 times). We report the average accuracy and standard deviation of each model on each test set in Figure 7.

Accuracy	Baseline	Logits Learning	Distillation	Multi-channel
Test data 1	0.634	0.683	0.698	0.708
Test data 2	0.629	0.684	0.694	0.702
Test data 3	0.630	0.673	0.698	0.697
Test data 4	0.636	0.679	0.695	0.706
Test data 5	0.631	0.684	0.697	0.708
Test data 6	0.638	0.676	0.700	0.700
Mean	0.633	0.680	0.697	0.703
Std	0.003	0.004	0.002	0.004

Figure 7: Average Accuracy of Student Models under Three Learning Paradigms and Baseline

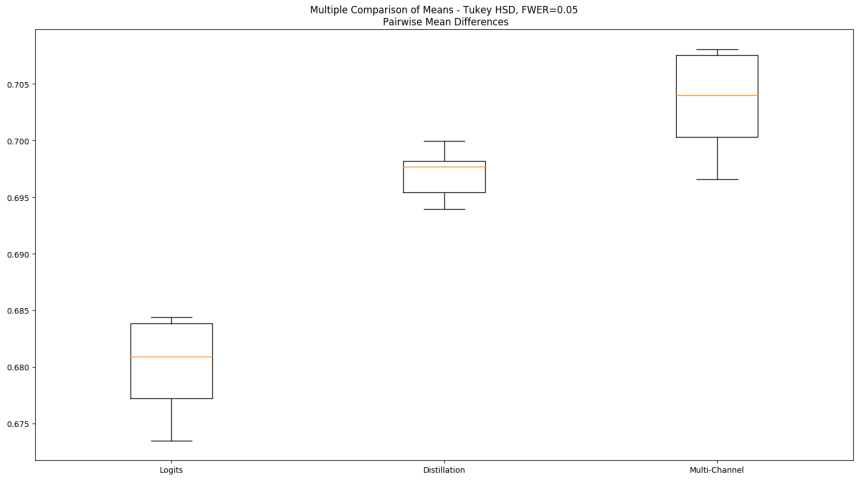


Figure 8: Performance of student models under three learning paradigms

The results shows two things: 1) all the teacher models' predictions are helpful to the student models and improve their performance by a big margin. 2) The performance of three student models under

three different learning paradigms are closed but still has some difference. In order to better measure the performance of the three teacher-student learning paradigms, we apply ANOVA test to evaluate the results. The null hypothesis is that "the three learning paradigms have no statistically significant difference with respect to their performances", and we want to test the alternative hypothesis and see if one learning paradigm is significantly better than the others.

One-way ANOVA Test: One way ANOVA is a technique that can be used to compare means of two or more samples. The reason why we don't choose multiple pair t test is because multiple significance test is more prone to have type I error. By using the module in Python we can easily obtain the p-value and F-value assosiated with F-distribution, which are F-value: 54.9707565915 and P-value: 1.24561472079e-07. We can see that the p-value are much smaller than the significant level $\alpha = 0.05$, therefore we can reject the null hypothesis, and conclude that the three learning paradigms are indeed different. We then perform the Tuckey post-hoc test, which is a moderate test that is less likely to make type I error, to pick the learning paradigm that outperforms the other. The post-hoc test results are shown in Figure 9.

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================
 group1   group2  meandiff lower  upper  reject
---------------------------------------------
Distilla  Logits   -0.017  -0.023 -0.011  True
Distilla Multi-Ch  0.0064  0.0004 0.0123  True
 Logits  Multi-Ch  0.0234  0.0174 0.0293  True
---------------------------------------------
```

Figure 9: Pair-wise test of performance under three different learning paradigms

The reason we don't use non-parametric method is because they are less powerful and they are more likely to make type I error, because they just use ranking instead of how different the results are to evaluate. Moreover, we can see from Figure 8 that all the results are very close to each other and therefore metrics like AUC might not be appropriate, so we just use parametric test to see if we should reject the null hypothesis.

According to the test results, we can see the three methods are indeed statistically significantly different, and we can rank the performance under three learning paradigms from high to low: Multi-channel label, Distillation, Logits learning. Therefore, we conclude that the Multi-channel labels method performs best on the specific teacher-student model-pair on the cifar10 dataset in this experiment.

6 Discussion

In this report we propose a new learning paradigm for model compression which is called multi-channel label method. By producing more semantic embedding of the data, the student can perform better than those using the distillation and logits learning methods, which are the two mainstream learning paradigms for model compression. Since we only compare these three learning paradigms on one teacher-student model pair and on a single dataset, there might be bias in the experiment results. But still, it offers us another way to do model compression and sheds some light on why the teacher's predictions are more helpful than the original labels to the student model. Figure 10 shows the semantic meaning of the multi-channel labels.

Future works on this direction can be how to learn better embedding (e.g. multi-channel label) than using convolution on the hidden representations. Also, attention-like mechanism can be incorporated in the scenario to weight each channel and it might help the student to focus on certain channels than the others. All in all, this report just provides another way of distillating knowledge from big complex models to shallow models and run some simple experiments to compare it with other common model compression methods, which I hope it can also be a little useful for unraveling the hidden mechanism of model compression.

Figure 10: Examples of multi-channel label and how they represent the similarity of different data. The multi-channel label learns something very similar to embedding which helps the model better discriminate the data.

7 References

[1]J. Ba and R. Caruana. Do deep nets really need to be deep? In Advances in neural information processing systems, pages 2654–2662, 2014.

[2]G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[3]A. Krizhevsky, V. Nair ,and G. Hinton."CIFAR-10 and CIFAR-100 datasets". https://www.cs.toronto.edu/ kriz/cifar.html.

[4]G. Hinton, O. Vinyals, and Jeff Dean. Dark Knowledge. http://www.ttic.edu/dl/dark14.pdf.

[5]L. Min, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013.

[6]D. Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. Machine learning 77.1: 103-123, 2009.

[7]Stefan van der Walt, S. Chris Colbert and Gael Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37.

[8]Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2017-12-09].

[9]Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[10]Qiao, Yu. THE MNIST DATABASE of handwritten digits. Retrieved 18 August 2013.