Udacity Robotics Software Nanodegree: Inference Project

Wisnu Prasetya Mulya

Abstract—This paper examines the inference performance of deep neural networks with different parameters: epoch and architecture. The models are trained using dataset that is collected by the Udacity's Robotics Software Nanodegree team and then the inference performances are compared. The results are all within Udacity's specification: minimum of 75% accuracy and maximum of 10ms inference time. The observation also suggests that the number of epochs and deep neural network architecture affects the performance of the inference process. Following the observation, another deep neural network is created using the similar parameters as the best performing network's and then an inference is conducted on the model.

Index Terms—Robot, IEEEtran, Udacity, LaTeX, deep learning.

1 Introduction

The problem of performance in a neural network is becoming more important as its application in the modern world is getting more prevalent. The increase in demand of embedded system specifically motivates the development of a better performing neural network in terms of accuracy and speed due to their limitations in areas such as connectivity and latency.

However, in optimizing performance, accuracy and inference time are found to be in a hyperbolic relationship [1]. This relationship implies that there is a trade-off in optimizing either one of them and it affects the decision making process of which neural network architecture to be implemented in a system, given certain minimum specifications and needs.

The work in this paper examines the performance of several deep neural networks with different parameters that would satisfy the specification needs set by Udacity: 75% minimum accuracy and 10ms maximum inference time. This condition would mimic the real-life situation of such needs and hence, would be proven applicable in the subsequent model that is created based on the best performing deep neural network.

The subsequent model would be trained using the data collected by the Author and it would have the task of identifying objects.

2 Background / Formulation

Two performance metrics are considered in the observation: accuracy and inference time. These two metrics would base the comparison between four different deep neural networks which vary in their parameter: epoch and architecture.

Epoch is the number of times a set of data is used to train a particular model and in this work two number of epochs are used: five and ten epochs. The architecture of a deep neural network is how a deep neural networks is layered and what type of neuron used in each layer. In this paper, there are two architecture networks considered: AlexNet [2] and GoogLeNet [3].

All networks are trained by using Nvidia DIGITS software and the performance are generated by the workspace provided in the Udacity's lesson page.

Further, another network is generated with parameters of the best performing network and then trained using another dataset collected by the Author. At the end, the performance of this network is evaluated by observing the result of an inference upon objects which have the same classification as the model training dataset's.

3 Data Acquisition

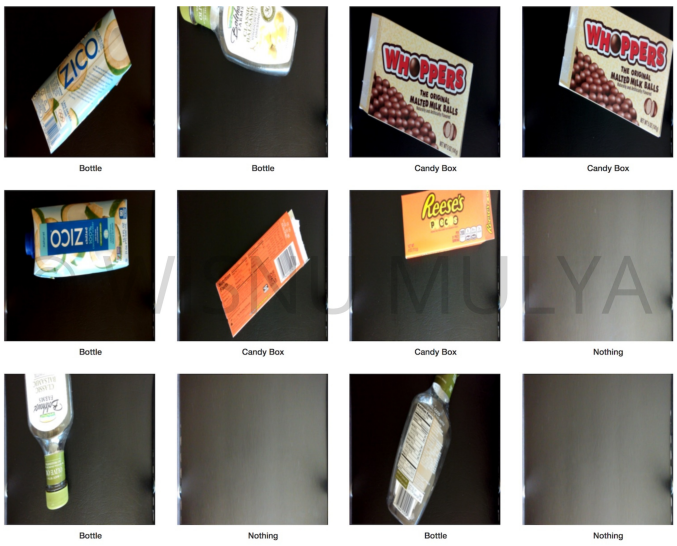

Fig. 1. Udacity Dataset

There are two datasets used in this work: Udacity's dataset and Author's dataset. The Udacity's is collected by the Udacity team and readily used by the Author, while the other one was collected by the Author using a MacBook Pro 15's installed camera.

Fig. 2. Author Dataset

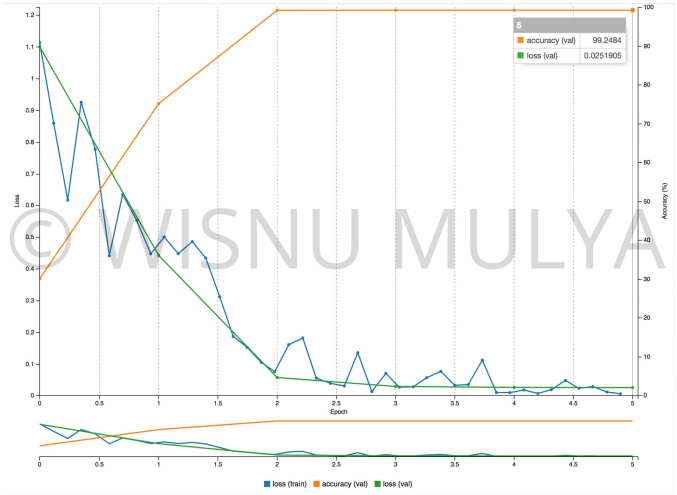

Fig. 3. AlexNet with 5 Epochs Training

The Udacity's dataset has three classification: nothing, bottle, and candy box, while the Author's has four classification: nothing, coin, pen, and phone. Both are in colors, but their sizes are different. The Udacity dataset's images have the dimension of 256x256 pixels, while the Author dataset's images have the dimension of 360x360 pixels.

In the aspect of background, the Udacity's dataset images have a conveyor belt background while the Author's have a black background and the objects are place on a white table.

Furthermore, the number of images collected is as follow:

- Nothing: 589 items
- Coin: 571 items
- Pen: 710 items
- Phone: 1,067 items

Out of the data collected, five images taken from each classes, except for the class "Pen" of which only four images taken, to perform the inference process, of whose result is shown on Figure 12.

4 Results

There are basically five results: four are the networks trained by the Udacity dataset and one is the network trained by the Author dataset.

4.1 Udacity Dataset: AlexNet with 5 Epochs

The AlexNet network with five epochs training results in validation rate of 99.2484% as seen in Figure 3. The following performance measurement of the inference process results in inference time of greater than 4.60ms and 75.4098% accuracy as seen in Figure 4.

4.2 Udacity Dataset: AlexNet with 10 Epochs

The subsequent training of AlexNet network is then conducted further to the total of ten epochs. The resulting performance, as seen in Figure 6, shows that three out of

```
root@cd5617c42e33:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20180201-112525-0ca5

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180201-112525-0ca5/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180201-112525-0ca5/snapshot_iter_300.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.66374 ms.
Average over 10 runs is 4.65132 ms.
Average over 10 runs is 4.63151 ms.
Average over 10 runs is 4.64238 ms.
Average over 10 runs is 4.64408 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14656  100 12340  100  2316   1034    194  0:00:11  0:00:11 --:--:--  2557

Your model accuacy is 75.4098360656 %
```

Fig. 4. AlexNet with 5 Epochs Evaluation

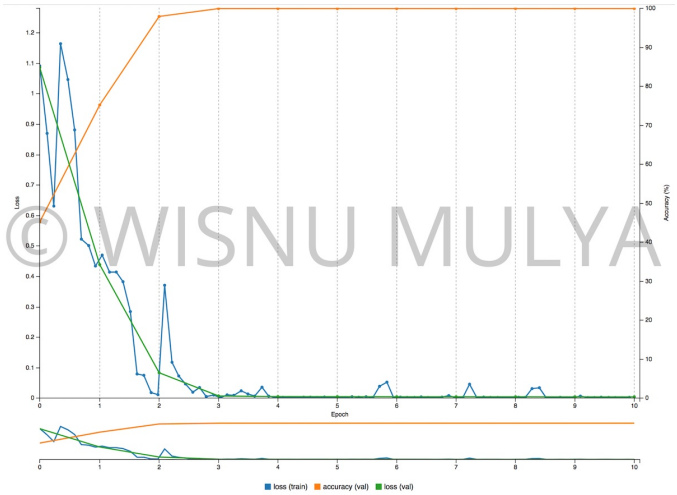

Fig. 5. AlexNet with 10 Epochs Training

```
root@cd5617c42e33:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20180201-111318-96a6

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180201-111318-96a6/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180201-111318-96a6/snapshot_iter_600.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.44272 ms.
Average over 10 runs is 4.44901 ms.
Average over 10 runs is 4.43308 ms.
Average over 10 runs is 4.15682 ms.
Average over 10 runs is 3.98496 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14629  100 12313  100  2316    963    181  0:00:12  0:00:12 --:--:--  2651

Your model accuacy is 75.4098360656_%
```

Fig. 6. AlexNet with 10 Epochs Evaluation

five inferences have inference time of greater than $4.4ms$, while one has $4.1ms$ and the another one has $3.9ms$. Finally, the resulting accuracy is 75.4098%.

4.3 Udacity Dataset: GoogLeNet with 5 Epochs

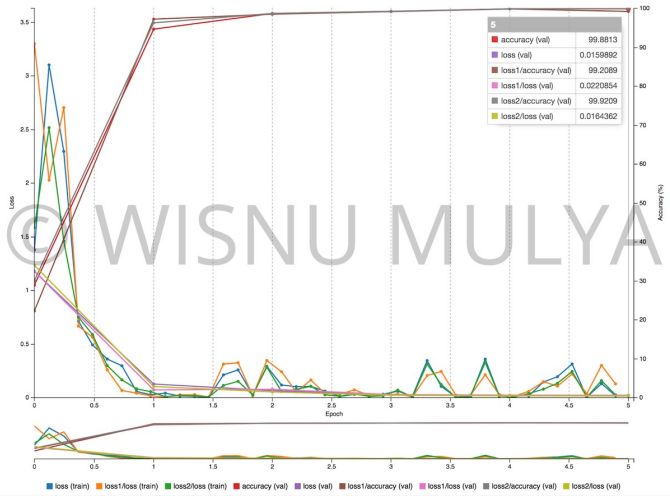

Fig. 7. GoogLeNet with 5 Epochs Training

Next, a different architecture network is trained by using the Udacity dataset: GoogLeNet. The first one is trained with five epochs and resulting in a 99.8813% validation rate as seen in Figure 7.

Subsequently, a measure in performance as seen in Figure 8 shows that the inference time are all above $5ms$, while its accuracy is 75.4098%.

4.4 Udacity Dataset: GoogLeNet with 10 Epochs

The final network trained using the Udacity data is with GoogLeNet architecture and it has ten epochs. Validation rate reaches 99.8022% as seen in Figure 9.

Further, the evaluation of the network results in $5.4ms$ in two out of five evaluations and $4.9ms$ for the rest, as seen in Figure 10.

```
root@cd5617c42e33:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20180201-113251-4398

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180201-113251-4398/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180201-113251-4398/snapshot_iter_1185.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.50591 ms.
Average over 10 runs is 5.42423 ms.
Average over 10 runs is 5.38532 ms.
Average over 10 runs is 5.39907 ms.
Average over 10 runs is 5.01775 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14665  100 12349  100  2316     171     32  0:01:12  0:01:11  0:00:01  2643

Your model accuacy is 75.4098360656_%
```

Fig. 8. GoogLeNet with 5 Epochs Evaluation

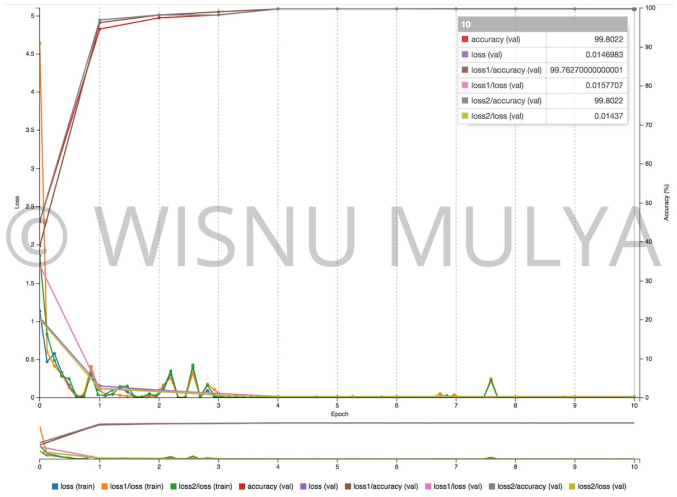

Fig. 9. GoogLeNet with 10 Epochs Training

```
root@84360a97f35d:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20180209-113811-1809

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180209-113811-1809/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180209-113811-1809/snapshot_iter_2370.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.43635 ms.
Average over 10 runs is 5.41723 ms.
Average over 10 runs is 4.94034 ms.
Average over 10 runs is 4.93047 ms.
Average over 10 runs is 4.93042 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14682  100 12366  100  2316     212     39  0:00:59  0:00:58  0:00:01  2444

Your model accuacy is 75.4098360656_%
```

Fig. 10. GoogLeNet with 10 Epochs Evaluation

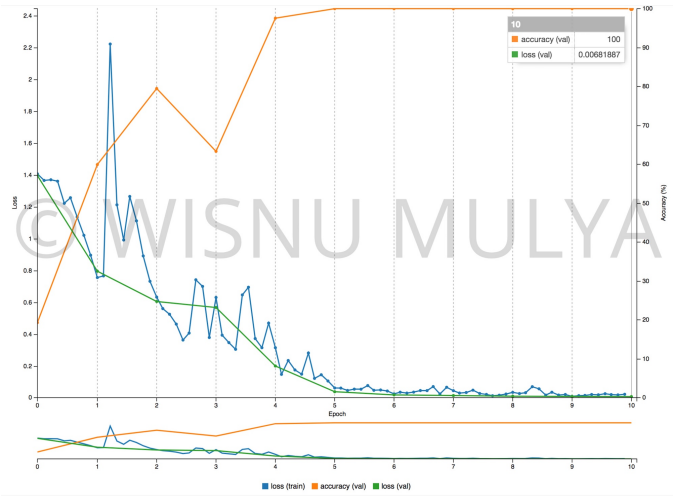

Fig. 11. Author's Data: AlexNet with 10 Epochs

All classifications

	Path	Top predictions							
1	/data/inference-data/normal_coin_286.png	Coin	98.35%	Pen	1.61%	Nothing	0.04%	Phone	0.0%
2	/data/inference-data/normal_coin_287.png	Coin	98.46%	Pen	1.5%	Nothing	0.04%	Phone	0.0%
3	/data/inference-data/normal_coin_306.png	Coin	99.76%	Pen	0.21%	Nothing	0.03%	Phone	0.0%
4	/data/inference-data/normal_coin_463.png	Coin	98.79%	Nothing	1.11%	Pen	0.1%	Phone	0.0%
5	/data/inference-data/normal_coin_571.png	Coin	88.38%	Pen	11.57%	Phone	0.03%	Nothing	0.01%
6	/data/inference-data/normal_nothing_585.png	Nothing	99.6%	Coin	0.4%	Pen	0.0%	Phone	0.0%
7	/data/inference-data/normal_nothing_586.png	Nothing	99.65%	Coin	0.35%	Pen	0.0%	Phone	0.0%
8	/data/inference-data/normal_nothing_587.png	Nothing	99.67%	Coin	0.33%	Pen	0.0%	Phone	0.0%
9	/data/inference-data/normal_nothing_588.png	Nothing	99.64%	Coin	0.36%	Pen	0.0%	Phone	0.0%
10	/data/inference-data/normal_nothing_589.png	Nothing	99.71%	Coin	0.29%	Pen	0.0%	Phone	0.0%
11	/data/inference-data/normal_pen_57.png	Pen	99.63%	Phone	0.37%	Coin	0.0%	Nothing	0.0%
12	/data/inference-data/normal_pen_191.png	Pen	99.76%	Phone	0.25%	Coin	0.0%	Nothing	0.0%
13	/data/inference-data/normal_pen_199.png	Pen	99.94%	Phone	0.04%	Coin	0.02%	Nothing	0.0%
14	/data/inference-data/normal_pen_522.png	Pen	99.92%	Phone	0.08%	Coin	0.0%	Nothing	0.0%
15	/data/inference-data/normal_phone_150.png	Phone	100.0%	Pen	0.0%	Coin	0.0%	Nothing	0.0%
16	/data/inference-data/normal_phone_275.png	Phone	100.0%	Pen	0.0%	Coin	0.0%	Nothing	0.0%
17	/data/inference-data/normal_phone_441.png	Phone	100.0%	Pen	0.0%	Coin	0.0%	Nothing	0.0%
18	/data/inference-data/normal_phone_538.png	Phone	100.0%	Pen	0.0%	Coin	0.0%	Nothing	0.0%
19	/data/inference-data/normal_phone_728.png	Phone	100.0%	Pen	0.0%	Coin	0.0%	Nothing	0.0%

Fig. 12. Author's Data: Inference Result

4.5 Author Dataset: AlexNet with 10 Epochs

By choosing the best performing network, the training of a model using that network's parameter is conducted with the dataset collected by the Author. The validation rate achieved 100% at the end of the training as seen in Figure 11.

Further, an inference process is executed using nineteen images of the same objects and the same background as the dataset's used to train the network. The result shows that all images are predicted accurately by the model as seen in Figure 12.

5 Discussion

The data suggests that the best performing network when trained using Udacity dataset is AlexNet with ten epochs. The defining parameter is the inference time, which is observed to be lower than $4.5ms$ in all evaluation instances, while the other networks are all performing with inference time of greater than $4.6ms$.

The observation shows that higher epochs produces a network with faster inference time. Both in AlexNet and GoogLeNet, the one with a higher epochs parameter produces a model with faster inference time.

The same comparison on inference time is also observed with different network architecture. AlexNet is observed to have faster inference time to GoogLeNet in both five and ten epochs.

A peculiarity is observed on the accuracy parameter evaluated on networks trained by Udacity dataset: the results are all of the same value of up to the tenth decimal digit. This might have been caused by a technical error in the Udacity workspace environment where the evaluation code exists.

As for the network trained using Author's dataset, the result suggests a very high chance of classifying objects accurately given the environment when collecting the train data is preserved: the background, the table, and the same objects.

The high accuracy that the network trained by Author's dataset might be explained by the small variety of poses collected in the train and inference data. Also, sharp differences of objects' shapes might contribute to the high prediction given by the network: the coin has circle shape, the phone has rectangular shape, and the pen has a line shape.

Furthermore, since the task of the neural network created using the Author's dataset is to identify objects on a controlled environment, the inference time performance would be secondary to the accuracy performance, due to that a faster neural network would jeopardize the accuracy performance and this is not desirable in order to produce a neural network that has a function to identify objects without any time constraint.

6 Conclusion / Future work

The results of the work in this paper suggests that higher epoch number contributes to a better performing network when measured in inference time. Also, AlexNet is observed to perform better in inference time compared to GoogLeNet.

A peculiarity is observed on the accuracy given by the evaluation conducted on Udacity's workspace. It is observed that all networks have the same value of accuracy up to the tenth decimal digit. This might be due to a technical error in the evaluation code and a future work of rectifying the code or evaluating the result in different platform would be necessary to obtain a conclusive result on the effect of epoch number and type of network architecture on the accuracy of the model.

Furthermore, the high prediction result observed in the network trained using the Author's dataset might be due to the small pose variation within each object class and the huge shape variation between each objects. Even though the result is promising, the model is not a commercially viable product, since it is trained in a highly controlled environment and the data collected have small variation and thus, it is not representative of the real world situation where variation is greater and environment would vary.

Further work on collecting a more diverse dataset with different background, different image type (black and white or inverted), and varying objects within the same class would increase the chance of having a commercially viable product that would do well in a real life situation.

REFERENCES

[1] A. P. E. C. Canziani, Alfredo, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.
[2] G. E. H. Alex Krizhevsky, Ilya Sutskever, "Imagenet classification with deep convolutional neural networks," 2012.
[3] Y. J. P. S. S. R. D. A. D. E. V. V. A. R. Christian Szegedy, Wei Liu, "Going deeper with convolution," *CVPR2015*, 2015.