COMP34212 Coursework Report

Hanmin Liu
UoM ID:10851873

April 18, 2024

1 Introduction

As the field of robotic evolves, the requirements for robots' perception of environment, decision-making and motor actions such as grasping, moving and precise manipulation has improved. Which leads to a revolution of the field by introducing the technique deep learning.

Deep learning is the science of training large artificial neural networks which controls vast amounts of parameters that are used on modelling complex functions that contributes to representations of state from raw, high-dimensional, multimodal sensor data commonly found in robotic systems. It stands out for not requiring manual adjust feature vectors from sensor data during design.[8]

However, while deep neural network (DNN) has present great enhancement on robotic area, it has also present challenges, including a great need for labelled data, expensive computational cost and the translation in-between training and applications.[1]

The review has a focus on the convolution neural network (CNN), which is a feed-forward neural network that learns feature engineering by itself. CNN is known widely by its applications on areas of image classification, recognition and natural language processing. The report explored the change in accuracy of image classification variance that lead by the change in different hyper parameters and CNN topology.

2 Deep Learning Models in Robotics

Deep learning has offered a range of methods that are suitable for various fields in robotic area, including convolutional neural network (CNN) for grid-like data processing such as images and sentences, recurrent neural network (RNN) for sequential data like language processing and speech recognition, long short-term memory network (LSTM) which is a upgrade version of RNN that is able to deal with long sequences, generative adversarial network (GAN) for simulating training environment , reinforcement learning (RL) for training robots to complete a goal, and etc. By utilizing the methods, deep learning has an influence on robotic field that is not ignorable with vast amount of practical examples. Take CNN and RL for example, Tesla Autopilot system is using CNN to process the large amount of image data describing the surrounding environment of vehicle contributing to the most advanced driver assistance system in the world. What is more, the current best Go player (including human and robotic player) in the world, AlphaGo, developed by DeepMind [3] is equipped with RL. However, deep learning models are suffering with computational demands problems. In the field of CNN, a bottleneck happened in 2014 with a discovery that with deeper convolution layers, the accuracy of model is not growing anymore. The problem was temporally solved by introducing a new model in CNN field called Resnet that allow the performance of model not dropping by introducing an additional step in-between layers namely identity that skip residual connections, which essentially are connections that contributes worse accuracy when going deeper.[4] Yet the performance of CNN is growing with minor improvements as the neural network goes deeper, due to which, the computational demands is growing fast, essentially becomes a barrier of deep learning. Deep learning also requiring large amount of data to train effectively, collecting which can be costly in both financial and time aspects, what is more, for different areas, it is significantly difficult for the deep learning models to generalize without retraining. Which are potential barriers for disseminate of deep learning.[7, 1, 8]

3 Methodology

CNN is a suitable deep learning model for classification problem that requires the network to provide given image(s)' classification out of a number of choice. It has a structure of three abstract layers: input layer that receives raw data, hidden layers essentially intermediate layers connected in a feed-forward way computing the features of data through convolution with a non-linear activation function while compressing features with pooling layer, finally output layer, also called fully-connect (FC) layer which often uses a softmax function to provide a probability distribution among various classes. By training the model, the weights and biases for the convolution filter and FC will be adjust and fit the problem with a higher accuracy of classification.[7] The dataset used in the experiments is CIFAR-10 due to its low resolution of images provides a manageable size that allows a fast experiment while a 32x32 image size is sufficient for training models. The CNN model used in the experiments is a four-layer network:

$$Input + Conv_1 + ReLU + Pool + Dropout$$

$$Conv_2 + ReLU + Pool$$

$$Dense_1 + Dropout + ReLU$$

$$Dense_2 + ReLU + Output$$

which is a modified version of LeNet-5[6], one of the earliest CNNs. Convolution layers have filter numbers of 6 and 16 since the small data size, with same filter size of 5x5 as well as padding. Pooling layers have kernel size of 2 and strides of 2, using max pooling due to its general better performance than average pooling, mainly caused by averaging counts in bad features.[2]

Before experiments, variables used in the experiments such as batch size were setup. Three experiments were carried out by changing hyperparameters including number, size and strides of convolution filter one at a time during 3 trainings, while controlling other parameters and topology of model the same. An additional experiment was carried out for comparing the performance of 7 layers network with 4 layers with a modified version of AlexNet[5]. After finding out the benificial hyperparameter settings for CNN, a optimized network with 7 layers was created base on the modification of AlexNet:

$$Input + Conv_1 + ReLU + Conv_2 + ReLU + Pool_1$$

$$Conv_3 + ReLU + Conv_4 + ReLU + Pool_2$$

$$Conv_5 + ReLU + Dense_1 + ReLU + Dropout + Dense_2 + ReLU + Output$$

The convolution layers have filter numbers 96, 256, 384, 384, 256 while FC layers both have 128 neurons base on a mature CNN namely AlexNet, with modifications on amount of neurons in FC layers since the original AlexNet has 4096 neurons in each FC layer, which cause a significant increament in the running time without major improvement on the accuracy.

4 Results

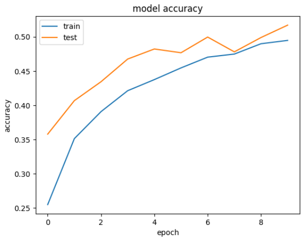

Figure 1: origin

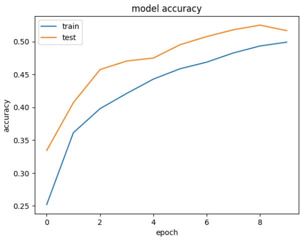

Figure 2: smaller kernel

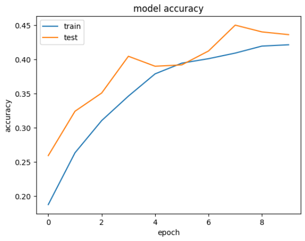

Figure 3: larger strides

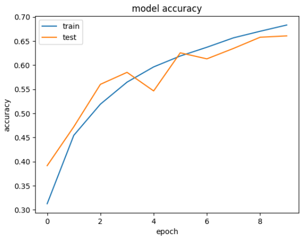

Figure 4: more filter

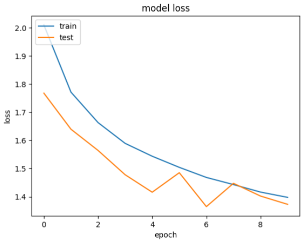

Figure 5: origin

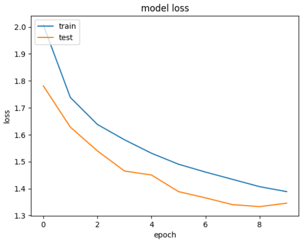

Figure 6: smaller kernel

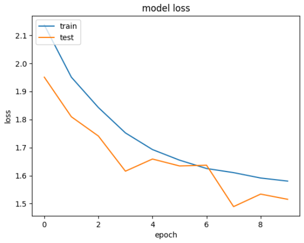

Figure 7: larger strides

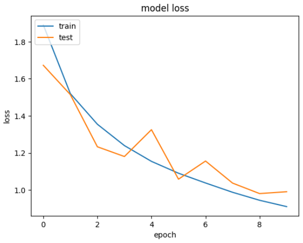

Figure 8: more filter

	origin	smaller kernel	larger strides	more filters
accuracy	0.5189	0.5199	0.4429	0.6547
score/loss	1.3727	1.3294	1.4951	0.9938

By observing the result of hyperparameters controlling experiments, compared to the origin network, the network with smaller kernel size has a minor increase on accuracy according to the table while testing accuracy is potentially more stable according to figure 1 and 2, the larger strides network has accuracy decreased by 7.6 percent with a more oscillating testing accuracy line, the network with more filters has a significant increase on accuracy by 13.58 percent. After the optimization, the test accuracy is able to reach 0.7835 with a validation accuracy of 0.9518. However the difference of validation and test accuracy indicates that the optimized network is overfitting the training data and requires more dropout layers to extract less features from training data.

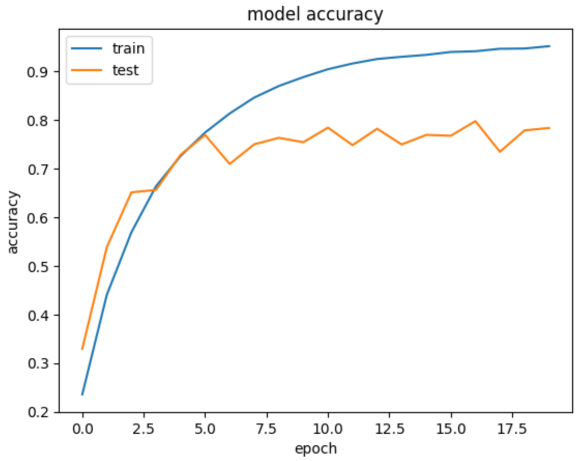

Figure 9: Optimized Accuracy

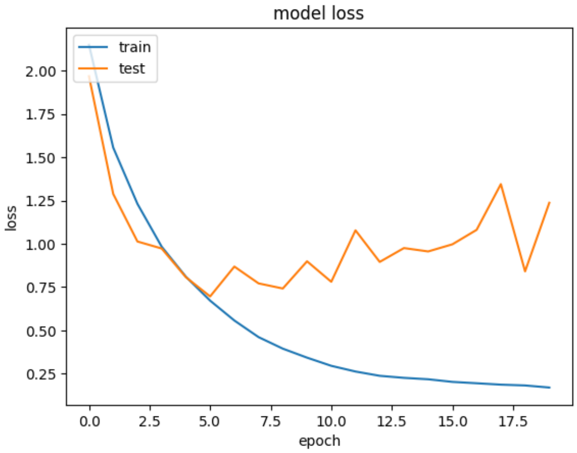

Figure 10: Optimized Loss

5 Conclusion

The observations on experiments with CNN on CIFAR-10 dataset lead to a conclusion that accuracy of network benifits from smaller kernel size and strides, as well as more filters and convolution/FC layers. By applying the founds on a optimized CNN model, the accuracy has improved 26.46 percents from 51.89 percents. This is reasonable since smaller kernel size and strides in combination of more filters essentially avoids data lost while providing more features and more convolution layers indicates more parameters the model can adjust hence a more detailed feature can be captured. However too much details can cause overfitting like the final model did, in this case a overfitting or using ResNet can help improving the model in the future work.

References

[1] Radouan Ait Mouha. “Deep Learning for Robotics”. In: *Journal of Data Analysis and Information Processing* 9.2 (2021).

[2] Dan Cireşan, Ueli Meier, and Juergen Schmidhuber. *Multi-column Deep Neural Networks for Image Classification*. 2012. arXiv: `1202.2745 [cs.CV]`.

[3] DeepMind. *AlphaGo*. Accessed: date-of-access. 2023. URL: `https://deepmind.google/technologies/alphago/`.

[4] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. arXiv: `1512.03385 [cs.CV]`.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems* 25 (2012).

[6] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: `10.1109/5.726791`.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (2015), pp. 436–444. DOI: `10.1038/nature14539`. URL: `http://dx.doi.org/10.1038/nature14539`.

[8] Harry A. Pierson and Michael S. Gashler. *Deep Learning in Robotics: A Review of Recent Research.* 2017. arXiv: `1707.07217 [cs.RO]`.

6 Appendix

Check experiment code at:
`https://drive.google.com/file/d/1rXiCrfx3d3_8m4dwGJxOyCy1-QlfVeoQ/view?usp=sharing`