DATA70132 Coursework: Neural Network Training for CIFAR-10

1. Introduction

This study is to try different hyperparameters of neural network training using CIFAR-10 dataset to find out the role of hyperparameters and to optimise the hyperparameters to get the best prediction performance. We can use celebrity CNNs such as ResNet to get high accuracy results, but a simple CNN shown in Figure 1 will be used as a starting point for this study. It gives room for improvement and makes it easier to see the effect of hyperparameter tuning.

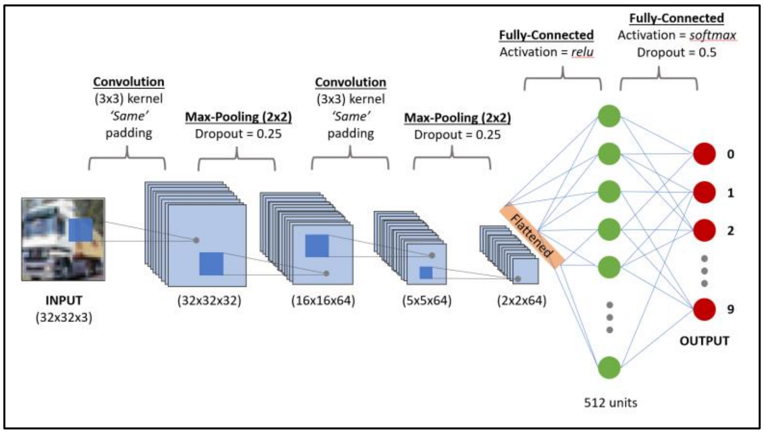

Figure 1. The network topology

Figure 2 below shows which hyperparameters will be tuned and in which order.

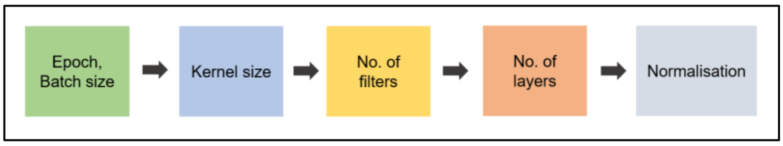

Figure 2. The hyperparameter tuning sequence

This study will test different epochs and batch sizes first to set an initial point for testing other time-consuming hyperparameters, such as the kernel size or the number of layers. If increasing the batch size or decreasing epoch can increase the accuracy or at least if it does not negatively affect the accuracy, testing other hyperparameters with less time will be possible. When testing the kernel size, the number of filters, and the number of layers, the research by Sinha et al. (2017) will be referenced to set the direction for tuning. After these, we will test if we can still improve the model by introducing normalisation into the model.

2. Simulation of different hyperparameters

2.1. Epoch, batch size

As Keskar et al. (2017) mentioned, increasing batch size may lead to degradation in the model quality. However, in this study, decreasing the batch size from 128 to 64 rather decreased accuracy

(0.7046->0.6828). Meanwhile, figure 3 shows that training over 29 epochs may lead to overfitting. Therefore, we set the batch size=64 and epoch=29.

2.2. Kernel size

As aforementioned, the research by Sinha et al. (2017) acts as a guide for hyperparameter tuning in this study. They did a grid search to get the best hyperparameters of 81.00 CIFAR-10 accuracy, using input layer=(16x16x3, 28x28x3, 32x32x3, 64x64x3), kernel size=(3x3, 5x5, 7x7), number of filters=(28, 32), number of layers=(11,13). The best model consists of input layer=32x32x3, kernel size=5x5, 28 filters, and 13 layers. Their results by grid search show that kernel size=5x5 works better than 3x3 or 7x7, 28 filters than 32 filters, and 13 layers than 11 layers. In this study, we could get model improvement by gradually increasing the kernel size from 3x3, 4x4 to 5x5 (Figure 4).

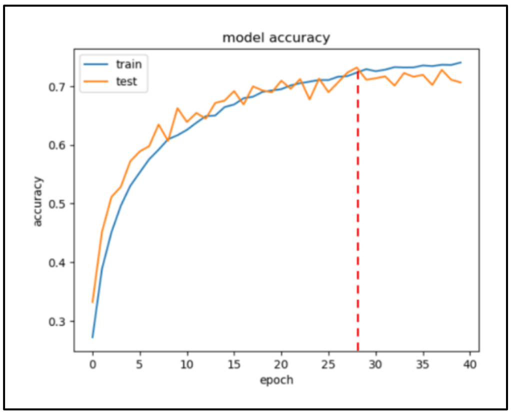

Figure 3. Model accuracy and epoch

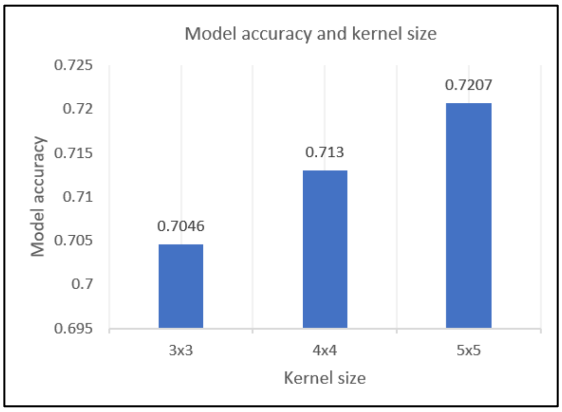

Figure 4. Model accuracy and kernel size

It appears that the effect of changing kernel size can differ from case to case when referring to the research by Ahmed et al. in 2020. They wrote that increasing the kernel size can also increase the time for training, but 3x3 filter showed better performance than 5x5 or 7x7 filter. Further research is needed to explore more of the relationship between model accuracy and filter size.

2.3. No. of filters

A filter size of 32 showed a better result than 28 (0.7207>0.7040), unlike Sinha et al. (2017). This appears to be because of differences in model structure or other hyperparameters. Ahmed et al. (2020) mentioned that 128 filters showed better results than 64 or 32 filters. Further research is required to generalise the relationship between model accuracy and the number of filters.

2.4. No. of layers

As Tuba et al. (2021) mentioned, adding more layers usually improves model performance but also increases the risk of overfitting. In this study, adding one more layer increased the model accuracy (0.7207->0.7391). It seems that the optimal number of layers can differ from case to case, depending on the difficulty of the task or the other model structure characteristics. Gong et al. (2022) showed that their improved ResNet10 model showed higher accuracy than ResNet101 (0.962>0.948) for remote sensing image classification.

2.5. Normalisation

Normalising the data further increased the accuracy (0.7391->0.8160). The improvement was bigger than any other hyperparameter tuning aforementioned. As Schilling (2016) mentioned, (batch) normalisation makes it possible to get higher accuracies on all datasets because of its regularising effect and more stable gradient propagation.

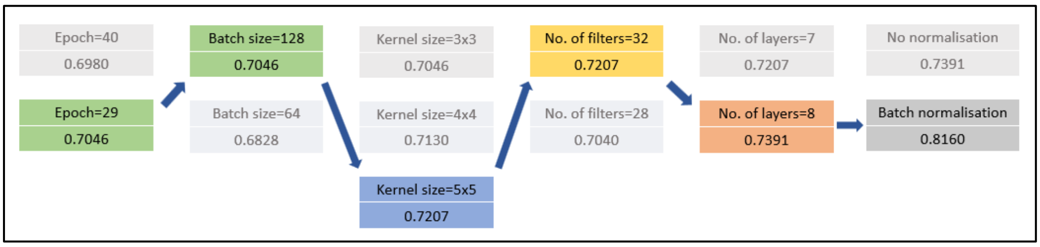

Figure 5. Chosen hyperparameters and model accuracy

3. Conclusion

Increasing the kernel size, the number of layers, the number of filters, and batch size led to higher accuracy of this CIFAR-10 classification model (Figure 5). Batch normalisation improved the model the most in this study. However, as mentioned above, the relationship between the model accuracy and hyperparameters cannot be generalised without further research because it depends on the task and the relationship between hyperparameters.

Furthermore, this study explores possible combinations of hyperparameters by inspirations from initial tests and other literature. As Tuba et al. (2021) mentioned, this way can be referred to as the 'guestimating' method. This was mainly due to time and computational power limitations. Further studies using well-known algorithms such as grid search or particle swarm optimisation are needed to test more hyperparameters and combinations of them than this study.

REFERENCES

- Ahmed, W. and Karim, A. (2020). 'The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN'. *2020 International Conference on Computer Science and Software Engineering (CSASE)*, Duhok, Iraq, 2020, pp. 88-93.
- Gong, J., et al. (2022). ResNet10: A lightweight residual network for remote sensing image classification. *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Changsha, China, 2022, pp. 975-978.
- Keskar, N., et al. (2023). '*On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*', arXiv: /abs/1609.04836.
- Tuba, E., et al. (2021). 'Convolutional Neural Networks Hyperparameters Tuning', in Pap, E. (eds.) *Artificial Intelligence: Theory and Applications*. Cham: Springer, pp. 65-84
- Schilling, F. (2016). *The Effect of Batch Normalization on Deep Convolutional Neural Networks.* Masters. KTH.