NEURAL NETWORKS & DEEP LEARNING
-
Report

1. The training methodology and chosen hyperparameters.

Epochs	Learning rate	Weight decay	Data transformation	Loss function	Optimizer	Batch size	Rate Scheduler
50	0.001	1e-4	Normalization, Random Horizontal Flip, Random Rotation	Cross Entropy Loss	Adam	64	(multi-step) Gamma = 0.5 Milestones = [30, 35, 40, 45, 50]

Figure 1. Table with final hyperparameters

The specifications of the code and methodology for the backbone, model, loss, optimizer and rate scheduler, training and model evaluation are explained in detail on the notebook. This document will focus on explaining the methodology and reasoning behind the chosen changes in the architecture and the final chosen hyperparameters, along with the curves for the evolution of the loss and the training and validation accuracies.

The model was trained by experimenting with different parameters and model architectures with aim of achieving the highest accuracy on the validation set. The starting values for the number of epochs, learning rate, weight decay, and batch size were based on already developed CNN architectures for the CIFAR10 dataset (Huang et al., 2017). Nevertheless, it was clear that these values needed adjusting for the specific CNN architecture used for this project. Now, each of the choices made to achieve a final validation accuracy of 0.8116 will be outlined.

The data augmentation techniques used on the CIFAR10 dataset achieved using transforms were random horizontal flip and random rotation. These helped the model generalize on the test dataset. These techniques are called Regularization techniques and through them both the train and test data set are also normalized to make computation faster.

The final number of epochs chosen was 50. This decision was made due to that, from the 53rd epoch, the values for the training and validation accuracy did not show a great range of change, implying that 0.81 was the best accuracy the model could achieve. The model was run on GPU, and each epoch took around 40 minutes to complete. The final batch size chosen was 64, since it was found that smaller batch sizes performed better and likely contributed to reducing the variance in gradient estimates during training. Moreover, the larger batch size required more memory and computation to run.

With regard to the optimizer, the Adam optimizer was chosen because it has shown good performance on various deep learning tasks and is known for being computationally efficient (Kingma et al., 2015). The final learning rate was set to 0.001, which was found to be an optimal balance between convergence speed and model accuracy. Different learning rates caused the model to overfit and underfit. In addition, weight decay was applied as a regularization

technique to prevent the model from overfitting. The value chosen was 1e-4, which is a small value since the model was not very complex.

The methodology also included experimentation with various model architectures. This was done by trying out different numbers of blocks, convolutions per block (k), and sizes of feature maps. Ultimately, it was found that a 3-block architecture, with 3 convolutions each and a maximum value of k=3, performed the best. Other numbers for k were tried, but they caused the model to perform badly. This architecture seemed to strike a good balance between complexity and performance, allowing the model to learn from the data while avoiding overfitting.

Finally, to train the model, Cross Entropy Loss was used as the loss function. Cross Entropy Loss is commonly used for multiclass classification tasks (Bishop, 2006). A MultiStepLR scheduler was also used. This gradually reduces the learning rate at specific milestones. The final milestones chosen were [30, 35, 40, 45, 50], and the gamma value was set to 0.5. This scheduler helped the model converge faster and achieve better performance on the validation set.

The final hyperparameters chosen seem to be a good fit for the model, achieving a balance between computational efficiency, convergence speed, and accuracy. However, there is still scope for improvement since higher accuracies could be achieved.

2. Curves for the evolution of the loss and the training and validation accuracies

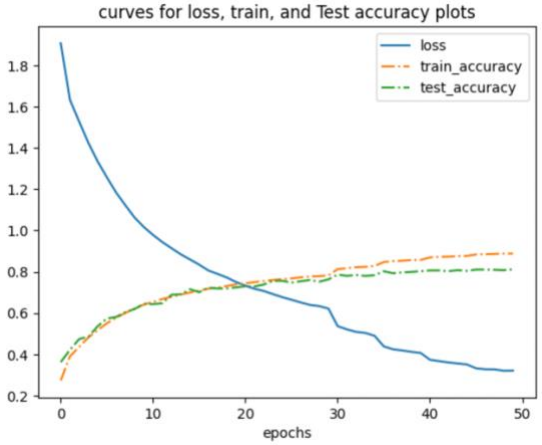

Figure 2. Curves for loss, train and test accuracy

The plot shows that the training and testing accuracies have a similar increasing pace, which implies that the model is not overfitting.

Moreover, the curve for the loss clearly shows the changes on the learning rate on the predefined milestones (starting at 30). These changes are less effective the closer the learning rate is to 0. The loss and train and test accuracies all seem to reach stagnation and not increase much after the 45th epoch. This implies that the model will not keep improving much after the 50th epoch.

Overall, the plot shows that the model seems to perform quite well.

Bibliography

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc.