

Introduction

CIFAR-10 is an established computer-vision dataset used for object recognition. It is a subset of the 80 million tiny images dataset and consists of 60,000 32x32 color images containing one of 10 object classes, with 6000 images per class. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

The aim of this project is to build a convolutional neural network model to classify images from the 10-classes dataset.

Methodology

The work mechanism was divided into several stages within the pipeline, where the following arrangement was adopted:

1. Data collection stage: The data was imported from the TensorFlow Datasets library, thus it will be taken into consideration as a Tensor Data object to work on in the next stages, it was divided into three groups, namely the training set, the validation set, and the test set, split into (40,000 samples), (10,000 samples) and (10,000 samples) respectively, with a total size of up to 60,000 samples.
2. After that comes the data exploration stage, where some samples were taken with the intention of displaying and exploring them, identifying the image characteristics, confirming their dimensions (32x32) and RGB channels.
3. Then now comes the role of the preprocessing stage, which is implicitly integrated under the pipeline framework and is applied to all data sets, while retaining the data augmentation option in the case of training data, where all previously defined processing methods are applied to each Tensor datasets.

4. For model building, the simple initial model was adopted in order to test it and explore the appropriate ceiling for this data, despite its simplicity, it gave ~ 65% validation accuracy, but it failed to capture complex patterns and its performance was not optimal, of course, other models and results will be discussed later.
5. The model training process included callbacks for the learning rate scheduler and early stopping, to ensure a better training process, especially since convolutional networks tend to be somewhat dynamic in the amount of learning rate that must decrease with the training process for better convergence, the validation set was used as a separate set for a more accurate evaluation of the test set later.
6. After training the model, the training results are displayed using the accuracy and test accuracy chart, and then the model is saved and reloaded for quick evaluation on the validation data.
7. In the final stage that completes the exploration processes and evaluation methods, the confusion matrix was adopted for advanced evaluation goals and exploring overlapping classifications and errors for each classification separately, and then advanced exploration of the nature of the model from displaying the filters in the first layer and displaying feature maps according to the layer index.

Results

Model	Accuracy	Loss	Val Accuracy	Val Loss	Learning Rate
Basic Model	0.6395	1.0290	0.6524	0.9913	1.0e-04
Complex Model (False Augmentation)	0.6480	1.0139	0.6690	0.9149	1.0e-07

Model	Accuracy	Loss	Val Accuracy	Val Loss	Learning Rate
More Complex Model (Augmentation with Batch Normalization & Dropout)	0.7964	0.5829	0.8377	0.4587	1.0e-08
Same as Before but no Batch/Dropout	0.7261	0.7720	0.7305	0.7599	1.0e-10
Best Model with 2 Conv layers per 1 pooling layer technique & Batch/Dropout	0.8136	0.5316	0.8521	0.4283	1.0e-07

**Learning Rates represent the final values at the last epoch, reduced dynamically by a learning rate scheduler from an initial value ($1e-4$).*

Discussion & Conclusion

The first model was not powerful enough and showed some overfitting,

So the easiest way to increase the complexity is to double the number of hidden layers and use data augmentation in addition to batch normalization and dropout layers between each layer, but the performance difference was negligible (about three percent improvement).

The main reason is that one convolution layer before the max pool was not enough for the model to complete the learning process at the layer level. Therefore, a two-layer and one max pool approach was adopted so that the model could learn the rest of the patterns without missing them. This actually led to a very large difference in the learning process even from the first 10 Epochs, as the performance of the validation test reached 83%.

Other experiments were conducted to study the effect of the improved layers on the training process, such as dispensing with batch and dropout, and a difference in performance loss of up to ten percent was observed.

As for data augmentation, it does not cause an increase in the model's performance, but rather reduces overfitting in the training data.

Finally, the model was improved by better tuning the layer parameters, especially batch and dropout, and then increasing the number of Epochs to

obtain the strongest possible performance with validation accuracy %**85.21**
and test accuracy **80.52%**.

By Ahmad Hudhud